

# TD 1

exercice 01 :

exercice 02 :

- ① calculs .
  - ② le hachage extensible ( bien expliqué )
  - ③ à faire tout seul
  - ④ devoir à rendre ~~XXXXX~~
- Attention j'ai eu un petit doute au niveau de la question d , quand une page  est pleine et doit être dédoubleée

# TD 1

## exercice 01 :

### Exercice 1 : Répartition des enregistrements dans les blocs

Soit la table Employé ci-dessous, contenant 600.000 Employés.

*Employé (num, nom, adresse, telephone, département)*

On suppose que la taille de l'attribut *département* soit 40 octets, de l'attribut *nom* 100 octets, de l'attribut *adresse* 50 octets, *telephone* 20 octets et *num* 10 octets.

On suppose que les pages de la base ont 3000 octets.

1. En considérant le stockage non-étendu :

- Combien d'enregistrements peut-on stocker dans une page ?
- Combien de pages sont-elles nécessaires pour stocker la table Employé ?
- Supposons que la table est stockée dans le fichier F1, donnez l'adresse des enregistrements 100.000ème, 3ème, 25ème et 150.000ème.
- Si on considère le stockage étendu, combien de pages seront-elles nécessaires pour stocker la table ?

$$a) 4 + 100 + 50 + 20 + 10 = 220$$

1 employé = 1 enregistrement

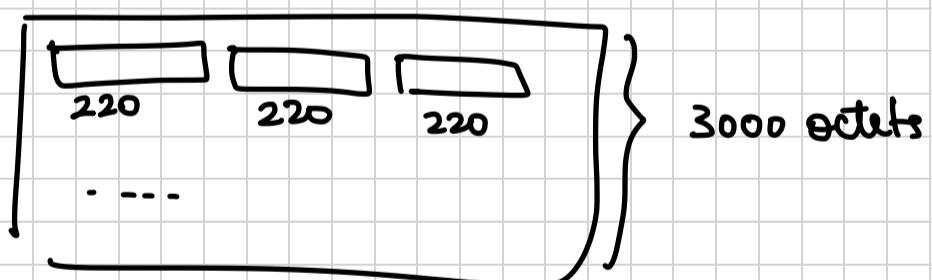
$$3000 \div 220 = 13, \dots = 13$$

car non étendu

$$b) 600\ 000 \div 13 = 46\ 153$$

pages

On a dû se tromper ici X



c) • 100 000 :

40      100      50      20      10

R-employés ( dept , Nom , adresse , Tel , num )

N- d'octets = 220

$$\frac{R}{B} = \frac{3000}{220} \approx 13$$

Adresse :

$$\rightarrow 100\ 000 : \frac{100\ 000}{13} + 1 = 7693$$

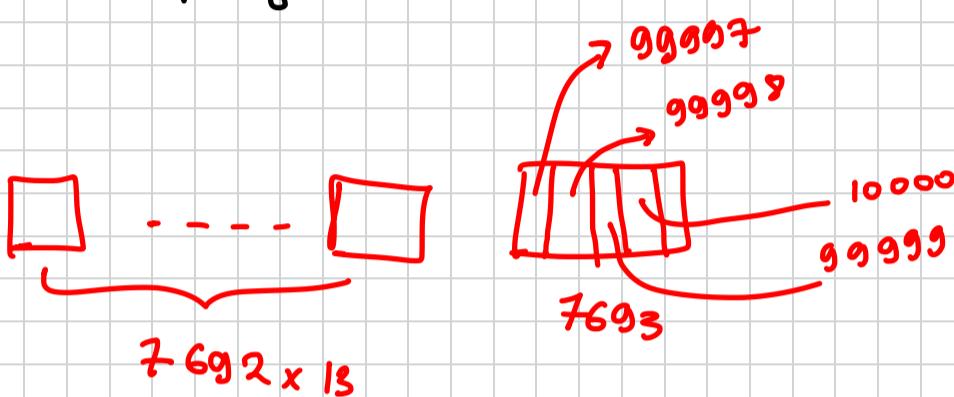
dans  
cette page là se  
trouve l'enregistrement

$$(7692 \times 13) + 1 = 99\ 997$$

- $7693 \times 13 = 100\ 009$

- $(100\ 000 - 99\ 997) + 1 = 3 + 1 = 4$

$\rightarrow 4^{\text{e}}$  page



- 3<sup>e</sup>me

$$\text{Page} = \frac{3}{13} + 1 = 1^{\text{ère}} \text{ page (il reste 3 enregistrements)} \\ 3^{\text{e}} \text{ enregistrement}$$

- 25<sup>ème</sup>

$$\text{Page} = \frac{25}{13} + 1 = 2^{\text{e}} \text{ page (reste 12)} \\ (12 \text{ enregistrement})$$

d) Si on considère le stockage étendu :

$$\frac{600\ 000}{3000} \times 220 = 44\ 000 \text{ pages.}$$

my try :

### Exercice 1 : Répartition des enregistrements dans les blocs

Soit la table Employé ci-dessous, contenant 600.000 Employés.

*Employé (num, nom, adresse, téléphone, département)*

On suppose que la taille de l'attribut *département* soit 40 octets, de l'attribut *nom* 100 octets, de l'attribut *adresse* 50 octets, *téléphone* 20 octets et *num* 10 octets.

On suppose que les pages de la base ont 3000 octets.

1. En considérant le stockage non-étendu :

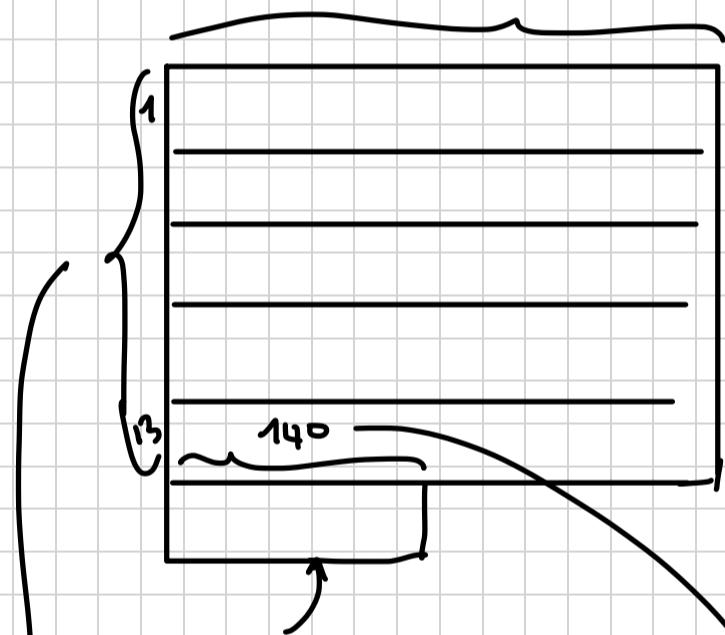
- ✓ a. Combien d'enregistrements peut-on stocker dans une page ?
- ✓ b. Combien de pages sont-elles nécessaires pour stocker la table Employé ?
- ✓ c. Supposons que la table est stockée dans le fichier F1, donnez l'adresse des enregistrements 100.000ème, 3ème, 25ème et 150.000ème.
- d. Si on considère le stockage étendu, combien de pages seront-elles nécessaires pour stocker la table ?

*fixé par le mc qui gère la base de données*

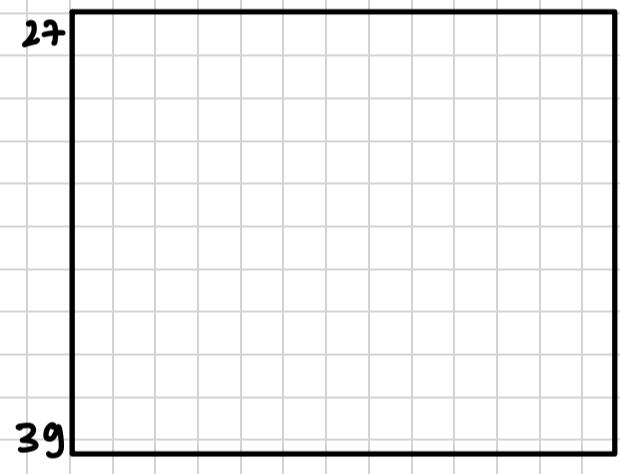
num	nom	adresse	téléphone	département
10	100	50	20	40

220 octets

= taille d'un enregistrement = 220 octets



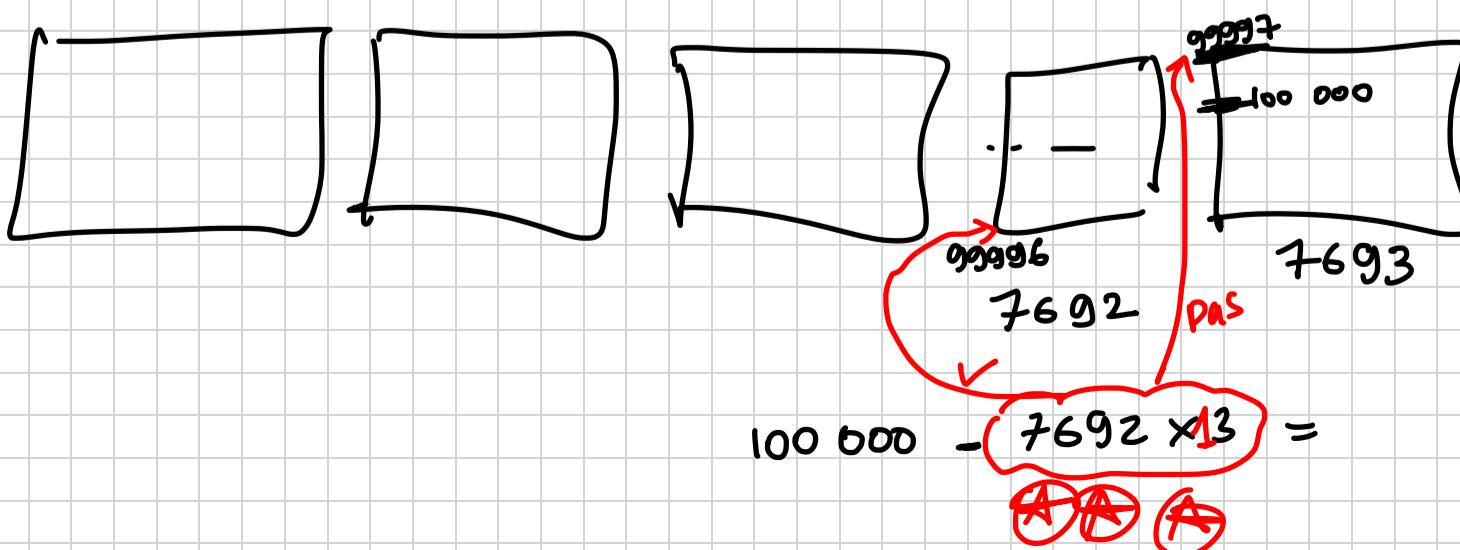
page = 3000 octets



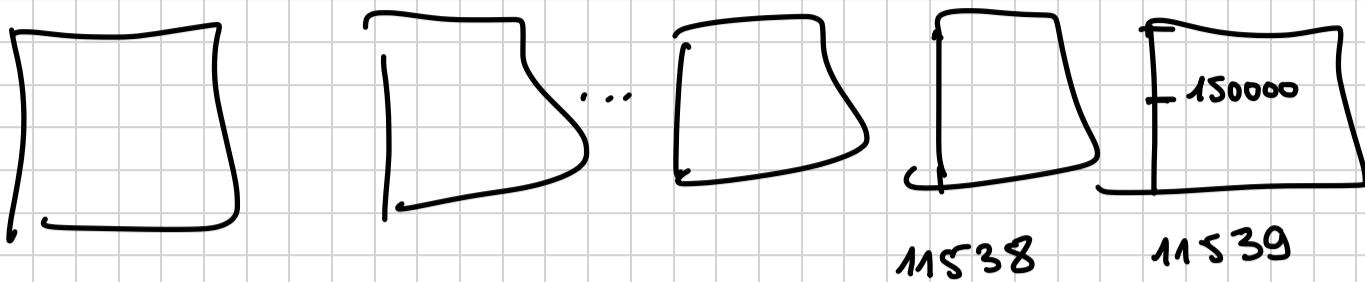
$$= 0,63 \dots \times 220 \text{ oct} \quad \text{et pas } \times 3000 \text{ attention}$$

$$\beta \text{ factor} = \frac{\text{nbv d'enregistrement}}{\text{page}} = \frac{3000}{220} = 13,63 = 13$$

$$\text{nombre de pages nécessaires: } \frac{600.000}{13} = 46153,8 = 46154 \text{ pages}$$



150 000



$$150000 - 13 \times 11538$$

f2. 11539 . 6

si on considère l'étendue :

$$\frac{600\ 000}{3000} \times 220 = 44000 \text{ pages}$$

## Exercice 02 :

### Exercice 2 : Organisation directe

1. Construisez 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 3 enregistrements par bloc et qu'on a 3 blocs primaires. Pour la gestion de collisions, utilisez la technique :

- a) Unchained overflow
- b) Chained overflow
- c) Multiple hashing.

chaque nombre est un overflow. ?

2. Construisez une structure de hachage extensible 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 3 enregistrements par bloc.

3. Décrivez l'évolution de la structure précédente (2) après les opérations suivantes :

- a) Suppression de l'enregistrement 11.
- b) Suppression de l'enregistrement 31.

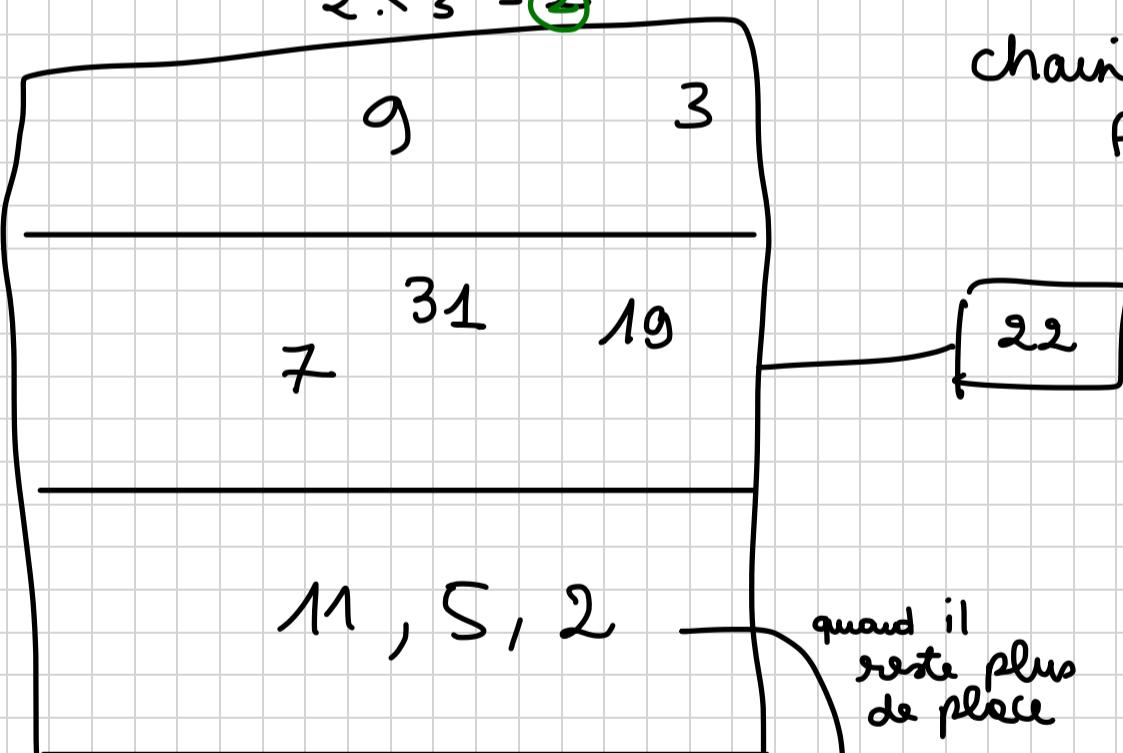
4. Soit la table ElementGenerique ci-dessous :

Create table ElementGenerique (  
id integer not null,  
nom,

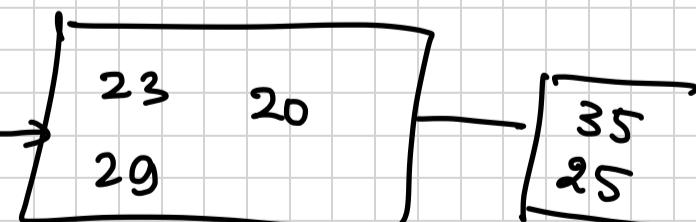
normale mt  
d'après ma  
compréhension  
ici enregistrent  
c'est juste le clé

### Exo 2.1 :

$$\begin{aligned} 3 \times 3 &= 0 \\ 19 \times 3 &= 1 \\ 11 \times 3 &= 2 \\ 5 \times 3 &= 2 \\ 2 \times 3 &= 2 \end{aligned}$$

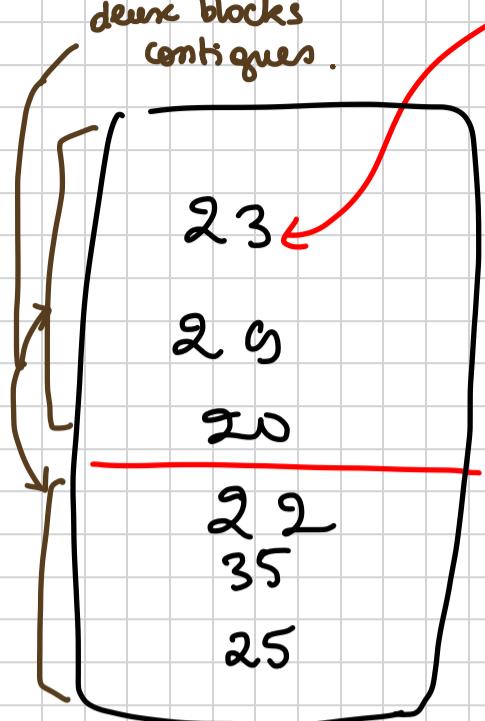


quand il reste plus de place



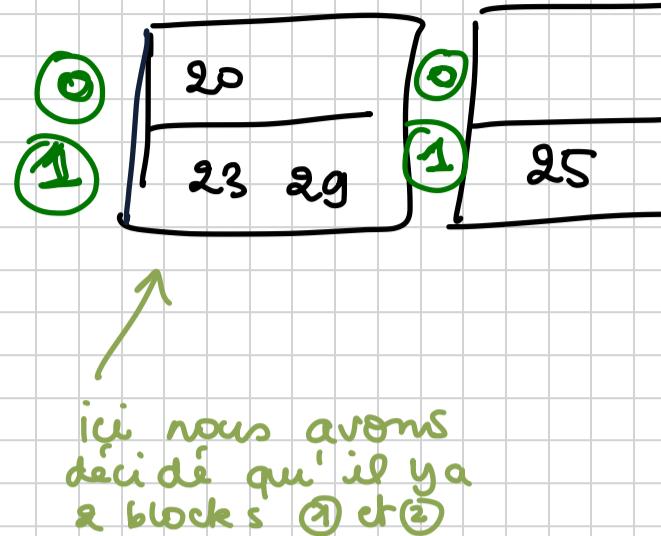
le fonction de hachage c'est modulo

## un chained over flow



23 est en overflow car  $23 \backslash 3 = 2$ .  
et le case 2 est déjà occupé  
par 3 valeurs  
3 enregistrement par page c'est  
pour ça on sépare

## multi heap hijacking



ici nous avons décidé qui il ya 2 blocks ① et ②

= 3 enregistrement

on a 3 blocks  $\Rightarrow$  donc la fonction de hashage  
c'est  $x \backslash 3$

Rq: types de gestion d'overflow

un enregistrement peut être en overflow dans ce

def:

cos il ya 3 méthodes de gestion des overflows:

- multi heap hijacking
- unchained overflow
- charged overflow

chaque méthode de gestion d'overflow a ses (+/-)

Rq: le nbre de blocks est défini au départ dans le hachage statique, et il ne peut pas changer car sinon on devra changer la table de hachage.

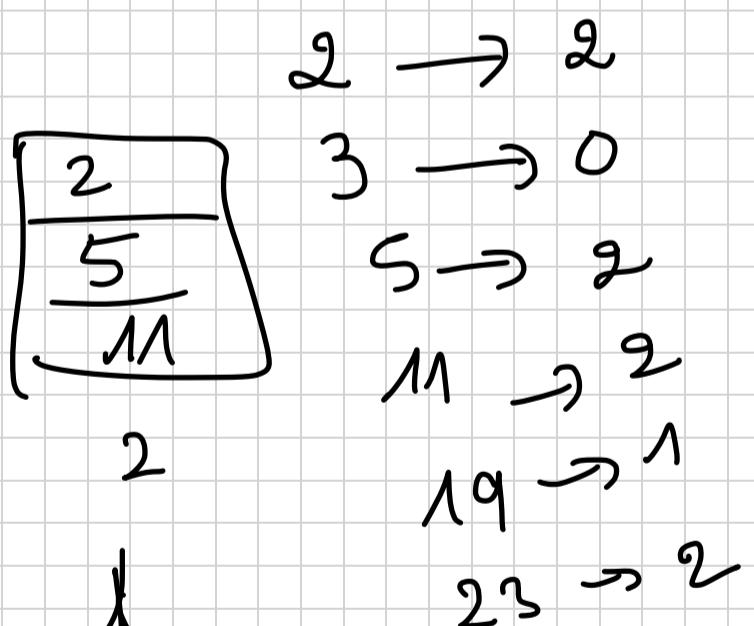
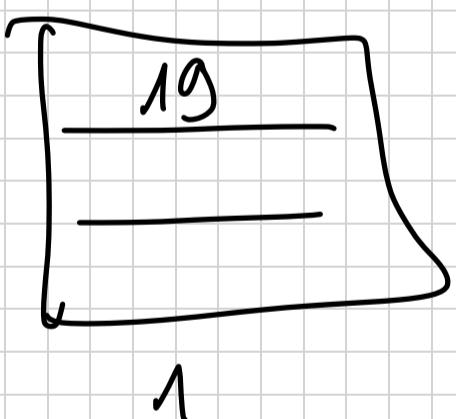
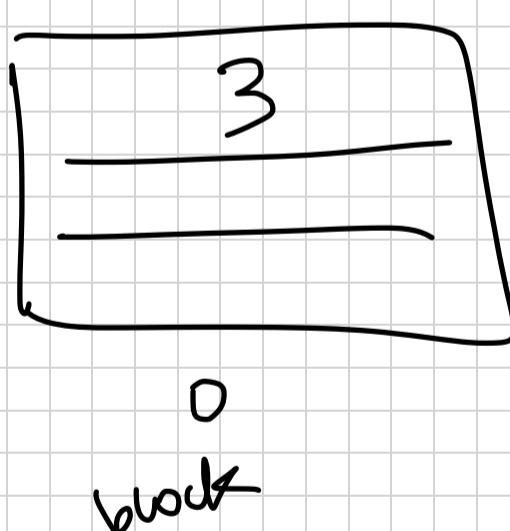
my try :

on a donc cette table : (je rajoute des info à la table pour comprendre)

l'indexation		
2	Maman	bNP
3	papa	LCI
5	tante	bNP
11	oncle	Aucun
19	cousine	:
23		

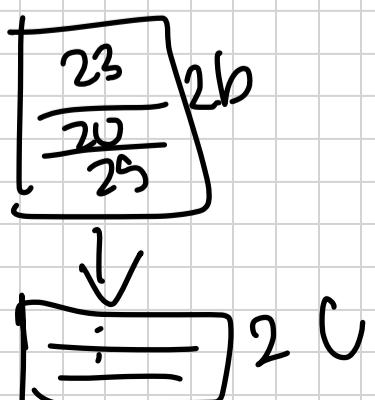
dans le méthodisé, le SGBD utilise une structure de hachage statique

il fait la structure de hachage avec ce quand une clé arrive



unchained  
over flow

Chained multi hashing



Nouvelle séance :

On fait la suite du feuille TD<sub>1</sub>

on avait fait l'exo : 2.1

on fait aujourd'hui 2.2 et 2.4

on laisse le 2.3 à nous de gref l'échir

### Exercice 2 : Organisation directe

1. Construisez 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 3 enregistrements par bloc et qu'on a 3 blocs primaires. Pour la gestion de collisions, utilisez la technique :
  - a) *Unchained overflow*
  - b) *Chained overflow*
  - c) *Multiple hashing.*
2. Construisez une structure de **hachage extensible** 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 3 enregistrements par bloc.
3. Décrivez l'évolution de la structure précédente (2) après les opérations suivantes :
  - a) Suppression de l'enregistrement 11.
  - b) Suppression de l'enregistrement 31.
4. Soit la table ElementGenerique ci-dessous :

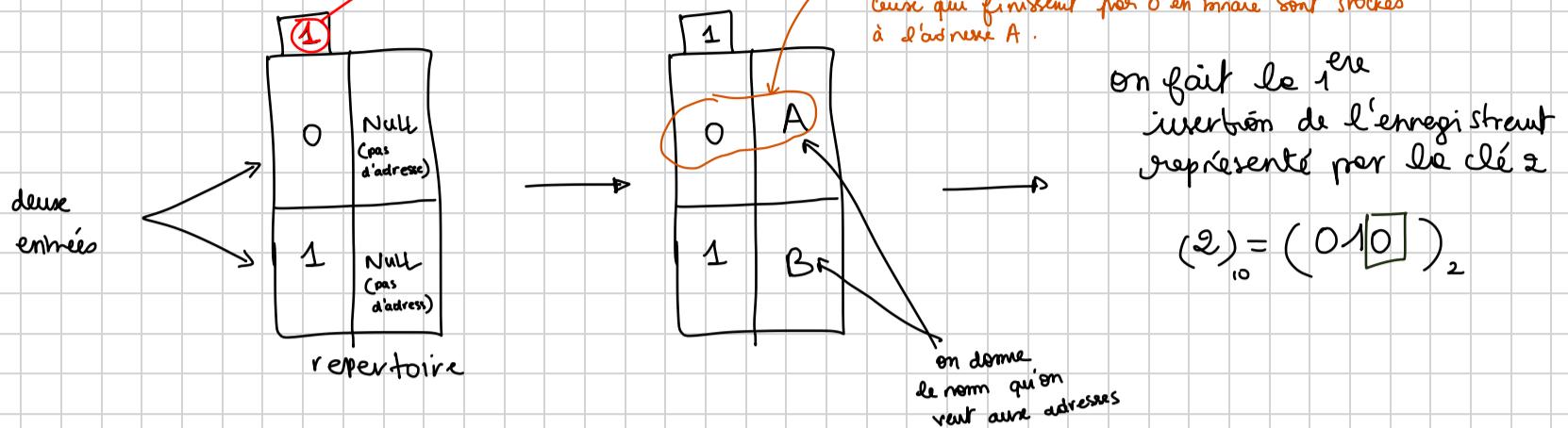
```
Create table ElementGenerique (
    id integer not null,
    nom,
    constraint pk_id primary key (id))
```

et les données :

```
(1,a);
(2,b);
(3,c);
(4,d);
(5,e);
```

② Structure de Hachage extensible

Rq: dans ces cases, on commence par un repertoire qui prend **un seul bit** donc dans le binaire je prend le dernier chiffre ex:  $5 = 0101$



on fait le 1<sup>er</sup> insertion de l'enregistrement représenté par la clé 2

$$(2)_{10} = (010)_2$$

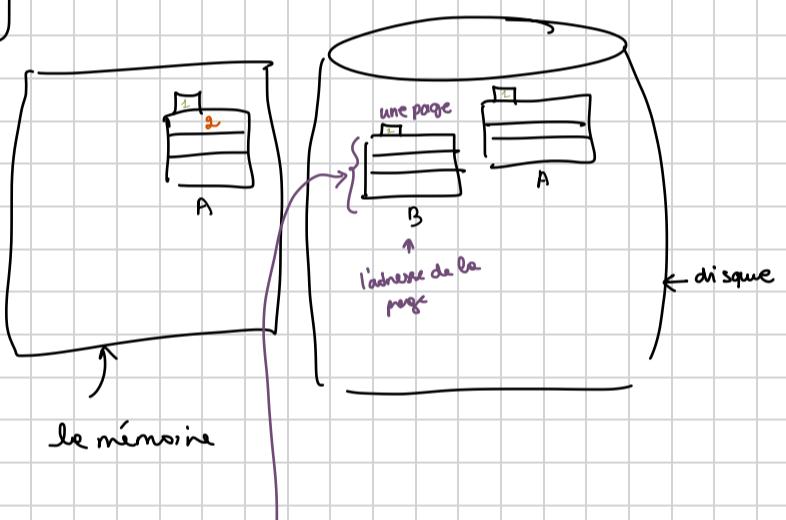
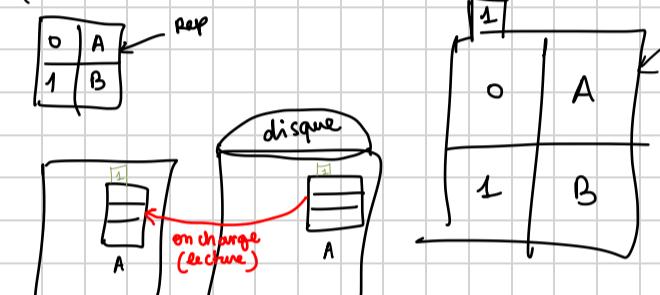
→ on considère le dernier bit (car on commence tjs par d=1)

c'est 0 donc on le mettra dans A car

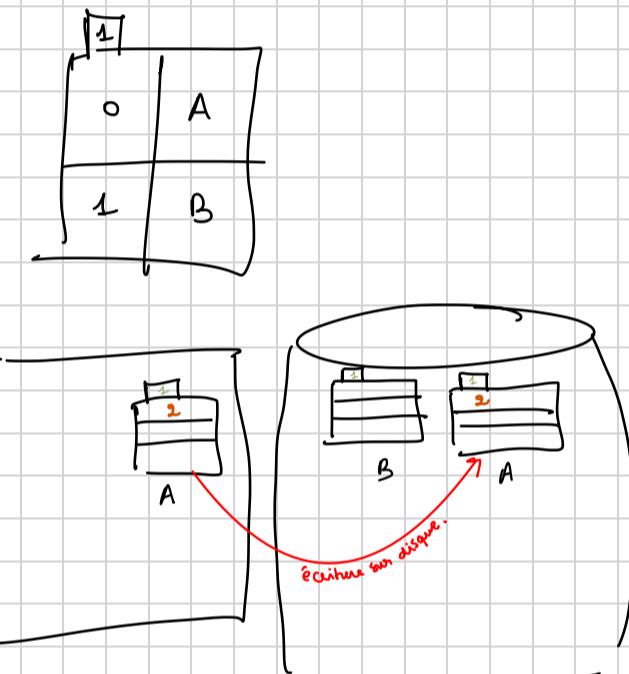
0	A
1	B

tout ce qui finit par 0 vont à l'adresse A  
et ceux qui finissent par 1 dans B

on charge le contenu de la page A depuis le disque vers la mémoire (en le lit eton le copie)



→ on écrit le 2 dans la page d'adresse A



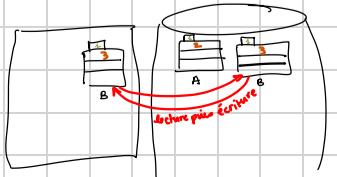
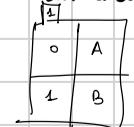
Rq: c'est indiqué dans l'énoncé que une page peut avoir 3 enregistrement max

→ on passe à l'enregistrement qui a la clé 3.

$$3 = 011 \Rightarrow \text{page } B$$

donc :

- on lit la page B du disque vers mémoire
- on voit qu'il reste de l'espace.
- on écrit l'enregistrement qui a la clé 3 dans la page B dans la mémoire
- on écrit sur le disque la page B (pour ajouter les nouvelles modifications)

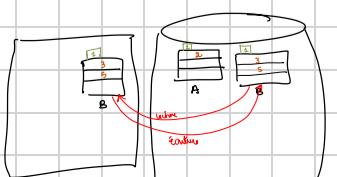
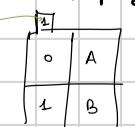


On passe à la clé 5

$$5 = 101$$

donc

- on lit la page B du disque vers mémoire
- on voit qu'il ya de la place.
- on écrit l'enregistrement qui a la clé 5 dans la page B dans la mémoire
- on écrit sur le disque la page B (pour ajouter les nouvelles modifications)



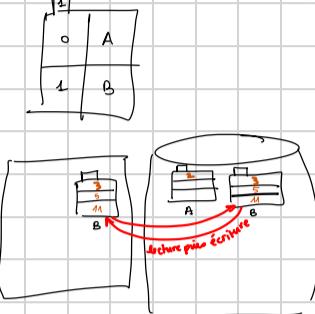
maintenant le page B est plein

on passe à l'enregistrement qui a la clé 11

$$11 = 1011 \Rightarrow \text{page } B$$

donc :

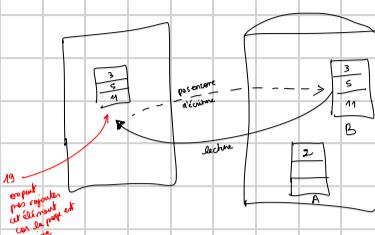
- on lit la page B du disque vers mémoire
- on voit qu'il reste de l'espace.
- on écrit l'enregistrement qui a la clé 3 dans la page B dans la mémoire
- on écrit sur le disque la page B (pour ajouter les nouvelles modifications)



on passe à l'enregistrement qui a la clé 10

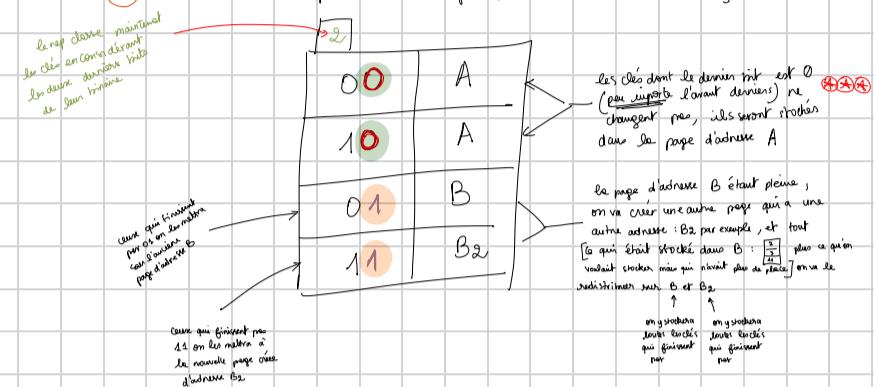
$$10 = 1001 \rightarrow \text{donc normalement en l'écrira sur la page } B$$

Le on lit la page B depuis le disque vers la mémoire et après le lecture on se rend compte qu'elle est pleine,



dans ce cas il faut un doubleage de la page d'adresse B

① étape 1 : le repertoire change, il devient comme ça :

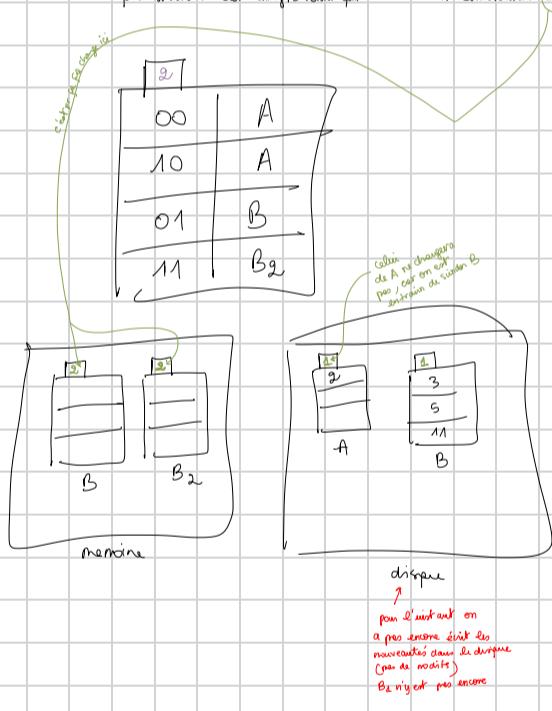


après que le repertoire soit modifié :

② étape 2 : on crée une nouvelle page d'adresse B2

et là donc B et B2 vont devenir des pages

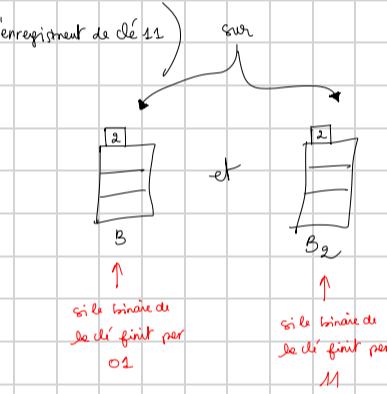
qui stockent des enregistrements qui en commanditant (2) dernier bit de la clé



Puis on réorganise le contenu de

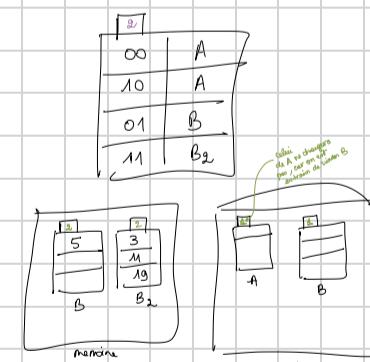


+ l'enregistrement de clé 11



$$\begin{array}{l} 3 = 0011 \rightarrow \text{sera stockée dans } B_2 \\ 5 = 0101 \rightarrow \text{"/ "/ } B \\ 11 = 1011 \rightarrow \text{"/ "/ } B_2 \\ 10 = 1001 \rightarrow \text{"/ "/ } B_2 \end{array}$$

Ce donne :



disque  
1

pour l'instant on

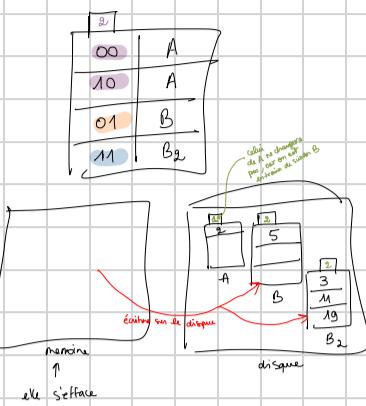
a pas encore écrit les

nouvelles données du disque

(pas de novità)

B2 n'a pas encore

Puis on écrit ce qui a été modifié  
sur le même dans le disque



→ on passe à l'enregistrement  
suivant :

$23 = 10111$  ← on regarde ses  
deux derniers car le répertoire organise selon  
les deux derniers Au plus

→ normalement selon le répertoire ça  
doit aller dans B2, mais quand  
on lit B2 on voit que c'est plein  
donc on doit dédoubler B2

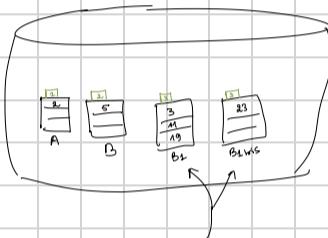
① le répertoire devient :

notez qu'on est en train de dédoubler B2 donc toutes les  
autres pages ne changent pas

1	2	3	4	5	6	7	8
0.00	A	-	-	-	-	-	-
1.00	A	-	-	-	-	-	-
0.10	A	-	-	-	-	-	-
1.10	B	-	-	-	-	-	-
0.01	B	-	-	-	-	-	-
1.01	B2	-	-	-	-	-	-
0.11	B2	-	-	-	-	-	-
1.11	B2bis	-	-	-	-	-	-

Technique de  
lecture par paquet  
de 4 octets

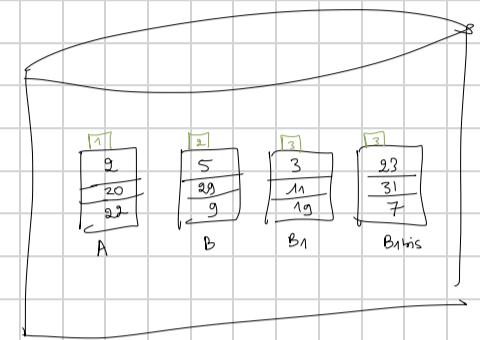
② le disque devient : Rq. J'en pas représenté la 8<sup>e</sup> page  
je suis d'accord avec un à disque



on redistribue 3, 19, 19 et 23

3 = 00111 → sur track dans B2.  
19 = 10011 → " " B1.  
19 = 10011 → " " B2.  
23 = 10111 → " " B2bis.

1	2	3	4	5
0.00	A	-	-	-
1.00	A	-	-	-
0.10	A	-	-	-
1.10	B	-	-	-
0.01	B	-	-	-
1.01	B2	-	-	-
0.11	B2	-	-	-
1.11	B2bis	-	-	-



## Corréction en Classe :

$2 = 010$ ✓	<table border="1"> <tr> <td>1</td> <td>0</td> <td>@500</td> </tr> <tr> <td>1</td> <td>1</td> <td>@A</td> </tr> </table>	1	0	@500	1	1	@A	<table border="1"> <tr> <td>1</td> <td>2</td> <td></td> </tr> </table> @500	1	2		2, 3, 5, 11, 19,																				
1	0	@500																														
1	1	@A																														
1	2																															
$3 = 011$	<table border="1"> <tr> <td>1</td> <td>0</td> <td>@500</td> </tr> <tr> <td>1</td> <td>1</td> <td>@A</td> </tr> </table>	1	0	@500	1	1	@A	<table border="1"> <tr> <td>1</td> <td>3</td> <td>5</td> <td>11</td> <td>@A</td> </tr> </table>	1	3	5	11	@A																			
1	0	@500																														
1	1	@A																														
1	3	5	11	@A																												
$5 = 101$ ✓	<table border="1"> <tr> <td>2</td> <td>00</td> <td>@500</td> </tr> <tr> <td>2</td> <td>01</td> <td>@A</td> </tr> <tr> <td>2</td> <td>10</td> <td>@500</td> </tr> <tr> <td>2</td> <td>11</td> <td>@100</td> </tr> </table>	2	00	@500	2	01	@A	2	10	@500	2	11	@100	<table border="1"> <tr> <td>1</td> <td>2</td> <td></td> </tr> </table> @500	1	2																
2	00	@500																														
2	01	@A																														
2	10	@500																														
2	11	@100																														
1	2																															
$11 = 1011$ ✓	<table border="1"> <tr> <td>2</td> <td>00</td> <td>@500</td> </tr> <tr> <td>2</td> <td>01</td> <td>@A</td> </tr> <tr> <td>2</td> <td>10</td> <td>@500</td> </tr> <tr> <td>2</td> <td>11</td> <td>@100</td> </tr> </table>	2	00	@500	2	01	@A	2	10	@500	2	11	@100	<table border="1"> <tr> <td>1</td> <td>5</td> <td>11</td> <td>19</td> <td>@100</td> </tr> </table>	1	5	11	19	@100													
2	00	@500																														
2	01	@A																														
2	10	@500																														
2	11	@100																														
1	5	11	19	@100																												
$19 = 10011$ ✓	<table border="1"> <tr> <td>2</td> <td>00</td> <td>@500</td> </tr> <tr> <td>2</td> <td>01</td> <td>@A</td> </tr> <tr> <td>2</td> <td>10</td> <td>@500</td> </tr> <tr> <td>2</td> <td>11</td> <td>@100</td> </tr> </table>	2	00	@500	2	01	@A	2	10	@500	2	11	@100	<table border="1"> <tr> <td>1</td> <td>2</td> <td></td> </tr> </table> @A	1	2																
2	00	@500																														
2	01	@A																														
2	10	@500																														
2	11	@100																														
1	2																															
$23 = 10111$ ✓	<table border="1"> <tr> <td>2</td> <td>00</td> <td>@500</td> </tr> <tr> <td>2</td> <td>01</td> <td>@A</td> </tr> <tr> <td>2</td> <td>10</td> <td>@500</td> </tr> <tr> <td>2</td> <td>11</td> <td>@100</td> </tr> </table>	2	00	@500	2	01	@A	2	10	@500	2	11	@100	<table border="1"> <tr> <td>1</td> <td>3</td> <td>11</td> <td>19</td> <td>@100</td> </tr> </table>	1	3	11	19	@100													
2	00	@500																														
2	01	@A																														
2	10	@500																														
2	11	@100																														
1	3	11	19	@100																												
$29 = 11101$	<table border="1"> <tr> <td>3</td> <td>000</td> <td>@500</td> </tr> <tr> <td>3</td> <td>001</td> <td>@A</td> </tr> <tr> <td>3</td> <td>010</td> <td>@500</td> </tr> <tr> <td>3</td> <td>011</td> <td>@100</td> </tr> <tr> <td>3</td> <td>100</td> <td>@500</td> </tr> <tr> <td>3</td> <td>101</td> <td>@A</td> </tr> <tr> <td>3</td> <td>110</td> <td>@500</td> </tr> <tr> <td>3</td> <td>111</td> <td>@104</td> </tr> </table>	3	000	@500	3	001	@A	3	010	@500	3	011	@100	3	100	@500	3	101	@A	3	110	@500	3	111	@104	<table border="1"> <tr> <td>1</td> <td>20</td> <td>22</td> <td></td> </tr> </table> @500	1	20	22			
3	000	@500																														
3	001	@A																														
3	010	@500																														
3	011	@100																														
3	100	@500																														
3	101	@A																														
3	110	@500																														
3	111	@104																														
1	20	22																														
$31 = 11111$	<table border="1"> <tr> <td>3</td> <td>000</td> <td>@500</td> </tr> <tr> <td>3</td> <td>001</td> <td>@A</td> </tr> <tr> <td>3</td> <td>010</td> <td>@500</td> </tr> <tr> <td>3</td> <td>011</td> <td>@100</td> </tr> <tr> <td>3</td> <td>100</td> <td>@500</td> </tr> <tr> <td>3</td> <td>101</td> <td>@A</td> </tr> <tr> <td>3</td> <td>110</td> <td>@500</td> </tr> <tr> <td>3</td> <td>111</td> <td>@104</td> </tr> </table>	3	000	@500	3	001	@A	3	010	@500	3	011	@100	3	100	@500	3	101	@A	3	110	@500	3	111	@104	<table border="1"> <tr> <td>1</td> <td>5</td> <td>29</td> <td>9</td> <td>@A</td> </tr> </table>	1	5	29	9	@A	
3	000	@500																														
3	001	@A																														
3	010	@500																														
3	011	@100																														
3	100	@500																														
3	101	@A																														
3	110	@500																														
3	111	@104																														
1	5	29	9	@A																												
$7 = 111$	<table border="1"> <tr> <td>3</td> <td>000</td> <td>@500</td> </tr> <tr> <td>3</td> <td>001</td> <td>@A</td> </tr> <tr> <td>3</td> <td>010</td> <td>@500</td> </tr> <tr> <td>3</td> <td>011</td> <td>@100</td> </tr> <tr> <td>3</td> <td>100</td> <td>@500</td> </tr> <tr> <td>3</td> <td>101</td> <td>@A</td> </tr> <tr> <td>3</td> <td>110</td> <td>@500</td> </tr> <tr> <td>3</td> <td>111</td> <td>@104</td> </tr> </table>	3	000	@500	3	001	@A	3	010	@500	3	011	@100	3	100	@500	3	101	@A	3	110	@500	3	111	@104	<table border="1"> <tr> <td>1</td> <td>3</td> <td>11</td> <td>19</td> <td>@100</td> </tr> </table>	1	3	11	19	@100	
3	000	@500																														
3	001	@A																														
3	010	@500																														
3	011	@100																														
3	100	@500																														
3	101	@A																														
3	110	@500																														
3	111	@104																														
1	3	11	19	@100																												
$9 = 1001$	<table border="1"> <tr> <td>3</td> <td>000</td> <td>@500</td> </tr> <tr> <td>3</td> <td>001</td> <td>@A</td> </tr> <tr> <td>3</td> <td>010</td> <td>@500</td> </tr> <tr> <td>3</td> <td>011</td> <td>@100</td> </tr> <tr> <td>3</td> <td>100</td> <td>@500</td> </tr> <tr> <td>3</td> <td>101</td> <td>@A</td> </tr> <tr> <td>3</td> <td>110</td> <td>@500</td> </tr> <tr> <td>3</td> <td>111</td> <td>@104</td> </tr> </table>	3	000	@500	3	001	@A	3	010	@500	3	011	@100	3	100	@500	3	101	@A	3	110	@500	3	111	@104	<table border="1"> <tr> <td>1</td> <td>20</td> <td>22</td> <td></td> </tr> </table> @500	1	20	22			
3	000	@500																														
3	001	@A																														
3	010	@500																														
3	011	@100																														
3	100	@500																														
3	101	@A																														
3	110	@500																														
3	111	@104																														
1	20	22																														
$20 = 10100$	<table border="1"> <tr> <td>3</td> <td>000</td> <td>@500</td> </tr> <tr> <td>3</td> <td>001</td> <td>@A</td> </tr> <tr> <td>3</td> <td>010</td> <td>@500</td> </tr> <tr> <td>3</td> <td>011</td> <td>@100</td> </tr> <tr> <td>3</td> <td>100</td> <td>@500</td> </tr> <tr> <td>3</td> <td>101</td> <td>@A</td> </tr> <tr> <td>3</td> <td>110</td> <td>@500</td> </tr> <tr> <td>3</td> <td>111</td> <td>@104</td> </tr> </table>	3	000	@500	3	001	@A	3	010	@500	3	011	@100	3	100	@500	3	101	@A	3	110	@500	3	111	@104	<table border="1"> <tr> <td>1</td> <td>5</td> <td>29</td> <td>9</td> <td>@A</td> </tr> </table>	1	5	29	9	@A	
3	000	@500																														
3	001	@A																														
3	010	@500																														
3	011	@100																														
3	100	@500																														
3	101	@A																														
3	110	@500																														
3	111	@104																														
1	5	29	9	@A																												
$22 = 10110$	<table border="1"> <tr> <td>3</td> <td>000</td> <td>@500</td> </tr> <tr> <td>3</td> <td>001</td> <td>@A</td> </tr> <tr> <td>3</td> <td>010</td> <td>@500</td> </tr> <tr> <td>3</td> <td>011</td> <td>@100</td> </tr> <tr> <td>3</td> <td>100</td> <td>@500</td> </tr> <tr> <td>3</td> <td>101</td> <td>@A</td> </tr> <tr> <td>3</td> <td>110</td> <td>@500</td> </tr> <tr> <td>3</td> <td>111</td> <td>@104</td> </tr> </table>	3	000	@500	3	001	@A	3	010	@500	3	011	@100	3	100	@500	3	101	@A	3	110	@500	3	111	@104	<table border="1"> <tr> <td>1</td> <td>3</td> <td>11</td> <td>19</td> <td>@100</td> </tr> </table>	1	3	11	19	@100	
3	000	@500																														
3	001	@A																														
3	010	@500																														
3	011	@100																														
3	100	@500																														
3	101	@A																														
3	110	@500																														
3	111	@104																														
1	3	11	19	@100																												

Exercice à rendre : feuille du TD1. Exercice 2 . question 4

## Element Générique

<u>id</u> (int)	nom (string)
1	a
2	b
3	c
4	d
5	e
6	f
7	g
8	h
9	i
10	j
11	k
12	l
13	m
14	n
15	o
16	p
17	q
18	r
19	s
20	t

une ligne = un enregistrement.

2.4.a : la taille d'un enregistrement =  $\Sigma$  des tailles des types des attributs

$$\text{La taille} = \text{size of (int)} + \text{size of (string)} = 8 + 32 = 40 \text{ octets}$$

2.4.b : le nombre d'enregistrements qu'on peut stocker par page = 
$$\frac{\text{taille d'une Page}}{\text{taille d'un enregistrement}}$$

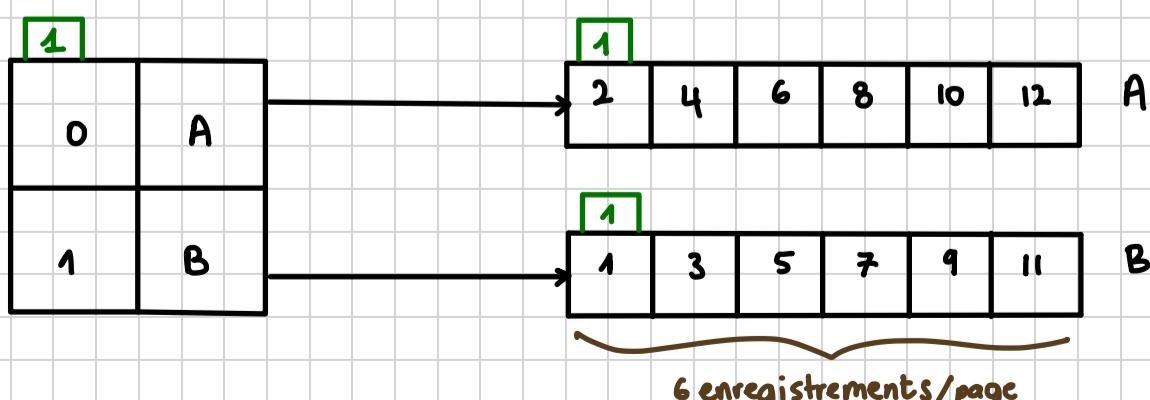
$$= \frac{240}{40} = 6 \text{ enregistrements /page}$$

## Rappel

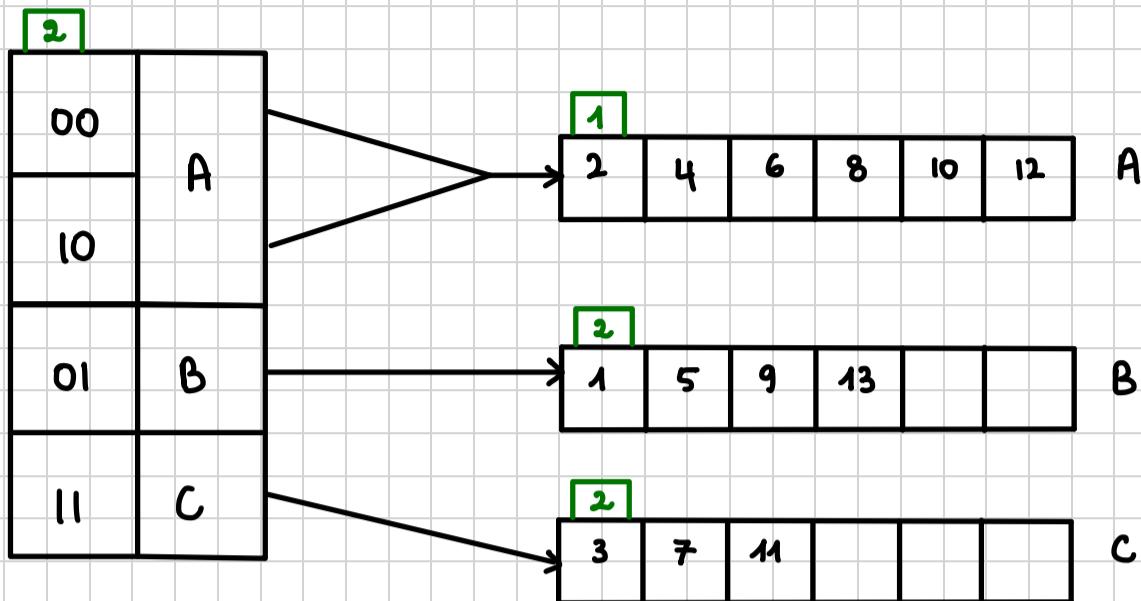
1	= 00001	11	= 01011
2	= 00010	12	= 01100
3	= 00011	13	= 01101
4	= 00100	14	= 01110
5	= 00101	15	= 01111
6	= 00110	16	= 10000
7	= 00111	17	= 10001
8	= 01000	18	= 10010
9	= 01001	19	= 10011
10	= 01010	20	= 10100

• on utilise une organisation de hachage extensible

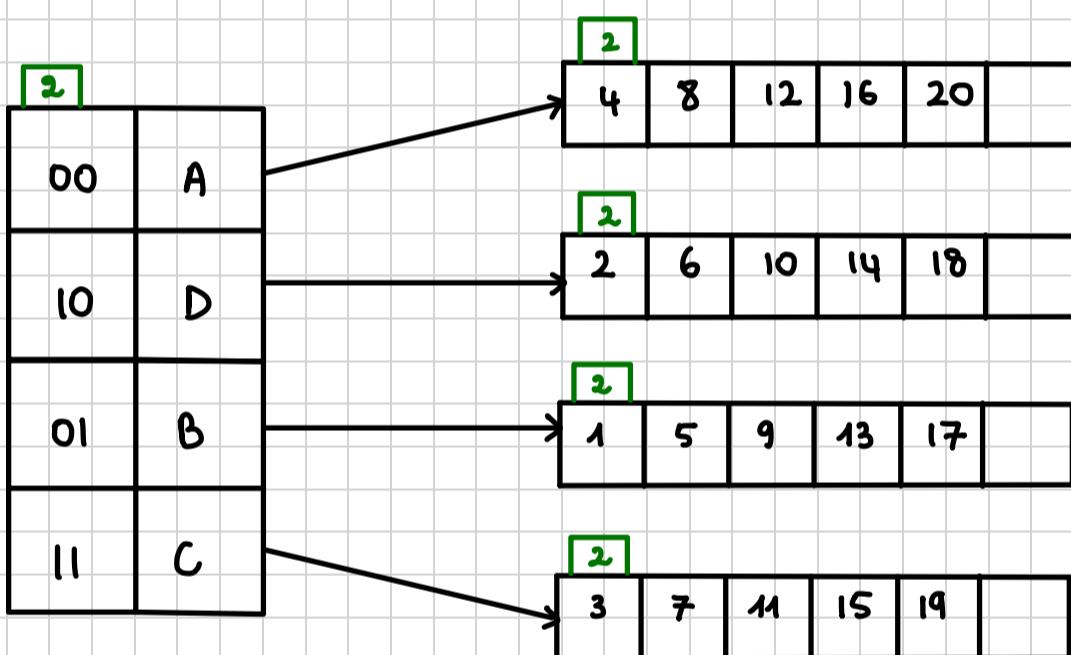
• le dernier bit des ids est utilisé pour déterminer l'adresse de la page

2.4.C : l'organisation de hachage après avoir inséré le 12<sup>e</sup> élément :

2.4.d : l'organisation de hachage après avoir inséré le 13<sup>e</sup> élément :



2.4.e : l'organisation de hachage après avoir inséré tous les éléments :



2.4.f : le nombre de Pages qu'ont été nécessaires pour stocker la table ElementGénérique :  
 les 4 Pages (A+B+C+D) + le répertoire

2.4.g : le nombre de pages qu'on doit lire pour répondre aux requêtes:

i) Pour localiser l'enregistrement de clé 7, on utilise le hachage extensible :  $7 = 00111$ , ses deux derniers bits (11) indiquent grâce au répertoire qu'il est dans la Page C (pas Nil)  
 ↳ donc : répertoire + Page C = 2 Pages

ii) il n'y a pas de hachage sur l'attribut nom, cela signifie que pour trouver le nom r, il faut lire toutes les Pages  
 ↳ donc : répertoire + toutes les 4 pages = 5 Pages

iii) l'enregistrement de clé 22 = 10110 doit être stocké dans la Page D (car ses deux derniers bits sont 10). il reste encore de l'espace dans la Page D donc pas de doublage nécessaire, on lit 2 fois puis on écrit  
 ↳ donc : le répertoire + Page D = 2 Pages

iv) les deux derniers bits de la clé 30 = 11110 sont : 10 donc cet élément devrait être dans la Page D  
 Pour supprimer l'élément il faut lire la Page D. Cependant, l'élément n'existe pas donc aucune suppression n'est effectuée.  
 ↳ donc : le répertoire + Page D = 2 Pages

# Réponses plus détaillées de ChatGPT pour mes révisions aux questions getf

## f) Combien de pages ont été nécessaires pour stocker la table ElementGenerique ?

D'après l'organisation que nous avons obtenue, voici la répartition finale des éléments dans les pages :

Page 1 (00) : ids [4, 8, 12, 16, 20] → 5 éléments  
Page 2 (01) : ids [1, 5, 9, 13, 17] → 5 éléments  
Page 3 (10) : ids [2, 6, 10, 14, 18] → 5 éléments  
Page 4 (11) : ids [3, 7, 11, 15, 19] → 5 éléments  
Donc, 4 pages sont nécessaires pour stocker les 20 éléments.

## g) Combien de pages doit-on lire pour répondre aux requêtes :

i. `Select * from ElementGenerique where id = 7`

Pour localiser l'élément avec `id = 7`, nous utilisons le hachage extensible :

Id 7 en binaire : 0111  
En utilisant les 2 derniers bits (11), on sait que cet élément est dans Page 4 (11).  
Résultat : 1 page à lire (Page 4).

ii. `Select * from ElementGenerique where nom = 'r'`

Il n'y a pas de hachage ou d'index sur la colonne `nom`. Cela signifie que pour trouver un élément basé sur son nom, il faut effectuer un `scan complet` des pages.

Résultat : 4 pages à lire.

iii. `INSERT INTO ElementGenerique (id, nom) VALUES (22, 'hf')`

L'id 22 en binaire est 10110. Les 2 derniers bits sont 10, ce qui signifie que cet élément doit être stocké dans Page 3 (10).

Page 3 contient actuellement 5 éléments (ids [2, 6, 10, 14, 18]), donc il y a encore de la place pour cet élément. Aucune extension ou réorganisation n'est nécessaire.  
Résultat : 1 page à lire (Page 3).

iv. `DELETE FROM ElementGenerique where id = 30`

L'id 30 en binaire est 11110. Les 2 derniers bits sont 10, donc cet élément devrait être dans Page 3 (10).

Cependant, l'élément avec `id = 30` n'existe pas dans cette page. Après vérification, aucune suppression n'est effectuée.  
Résultat : 1 page à lire (Page 3).

A  
Rq : chatGPT se trompe dans les réponses car il considère pas le répertoire comme Page  
La je copie les réponse juste pr avoir une explication claire avec des mots

**Remarque :** quand je pose des questions sur le hachage extensible il faut lui préciser qu'en utilise le dernier bit (0 ou 1) pour déterminer la page. sinon il se trompe

**Remarque 2 :** la Prof dit que lors du dédoublement (quand on veut ajouter un élément à une page pleine) il ne faut **absolument** pas toucher aux pages qu'on ne dédouble pas (elles restent inchangées)  
il ya juste dans le répertoire que plusieurs bits correspondent à la même adresse.

## Question

est-ce que le répertoire est une page ? ⇒ oui le répertoire est une page.

2.4.e :

3	
000	A
100	D
010	A
110	D
001	B
101	B
011	C
111	C

3							A
3							D
2	1	5	9	13	17		B
2	3	7	11	15	19		C

# Réponses drive : à confirmer

## Exercice 1

- 1) Index secondaire en Arbre B+ sur l'attribut prix du type (clé de recherche, adresse de l'enregistrement)
- 2)
  - Taille enregistrement : taille prix + taille d'une adresse = 30 + 8 = 38
  - Page =  $512 - 8 = 504$  octet pour les enregistrements,  $504 / 38 = 13$  enregistrements par page
  - Nombre d'enregistrements : 500 enregistrements
  - Nombre de page :  $500 / 13 = 38,46 \rightarrow 39$  pages
  - Nœud intermédiaire : 12 clé et 13 adresses
  - 2 niveau d'adresse + données
- 3)
  - A. Avec Index :
    - Meilleur des cas : 2 entrée + 3 pages + 34 enregistrements = 39
    - Pire des cas : 2 entrée + 4 pages + 34 enregistrements = 40
  - B. Sans Index
    - Taille enregistrements : 90
    - Enregistrements / page =  $512 / 90 = 5$
    - Nombre de page =  $500 / 5 = 100$
    - On a donc besoin de 100 pages
- 4)
  - A) Avec Index :
    - On ne peut pas car l'index n'est pas ordonné par rapport à l'id
  - B) Sans index :
    - On lit toute la table donc 100 pages

## Exercice 2

1)

Orde 1	
0	A
1	B

1	A	1,3,5,7,9,11
1	B	2,4,6,8,10,12

Orde 2	
00	A
01	B
10	A
11	C

2	A	1,5,9,13
1	B	2,4,6,8,10,12
2	C	3,7 ,11

Orde 2		
00	A	
01	B	
10	D	
11	C	

2	A	1,5,9,13,17
2	B	4,8,12,16,20
2	C	3,7 ,11,15,19
2	D	2,6,10,14,18

- 2) Taille enregistrements :  $32 + 8 = 40$  octet
- 3) Nombre d'enregistrements / page :  $240 / 40 = 6$
- 4) On a besoin de 4 pages + répertoire
- 5)
  - i) 2 pages – 1 repertoire et 1 pages
  - ii) 5 pages car on parcours tout
  - iii) 3 pages – 2 d'accès et 1 d'écriture
  - iv) 2 pages