

Cours de bases de données,  
aspects systèmes,  
<http://sys.bdpedia.fr>

Hachage extensible

# Hachage extensible

Le hachage extensible permet de réorganiser la table de hachage en fonction des insertions et suppressions.

Idée de base : la fonction de hachage  $h$  est fixe, **mais** on utilise les  $n$  premiers bits du résultat  $h(c)$  pour s'adapter à la taille de la collection.

Par rapport à la version de base du hachage, on ajoute deux contraintes.

- le nombre d'entrées dans le répertoire est une puissance de 2
- la fonction  $h$  donne toujours un entier sur 4 octets (32 bits)

La taille du répertoire tend à croître rapidement et peut poser problème.

## Exemple : le hachage des 16 films

On ne montre que le premier octet.

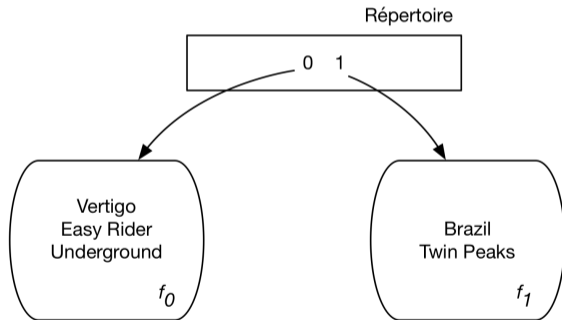
titre	$h(\text{titre})$
Vertigo	01110010
Brazil	10100101
Twin Peaks	11001011
Underground	01001001
Easy Rider	00100110
Psychose	01110011
Greystoke	10111001
Shining	11010011

La fonction  $h$  est immuable.

# Construction de la table

On départ on utilise seulement le premier bit de la fonction

- Deux valeurs possibles : 0 et 1
- Donc deux entrées, et deux fragments
- L'affectation d'un enregistrement dépend du premier bit de sa fonction de hachage



Avec les 5 premiers films.

# Insertions

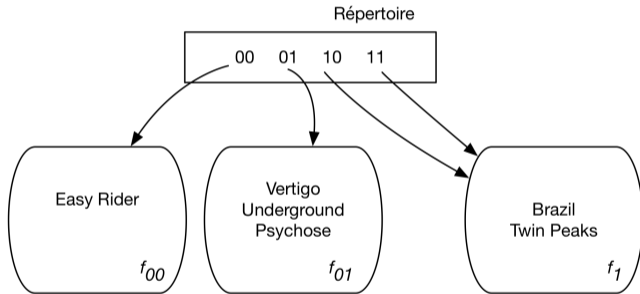
Supposons 3 films par fragment. L'insertion de Psychose (valeur 01110011) entraîne le débordement du premier fragment.

- On double la taille du répertoire : quatre entrées 00, 01, 10 et 11.
- On alloue un nouveau fragment pour l'entrée 01
- Les entrées 10 et 11 pointent sur le **même** fragment.

Le répertoire grandit, mais dans l'espace de stockage on ajoute seulement un nouveau fragment.

# Illustration

Les enregistrements anciennement stockés dans le fragment 0 sont répartis dans les fragments 00 et 01 en fonction de leur second bit.



Réorganisation **locale**, donc coût acceptable.

# Insertions suivantes

Deux cas.

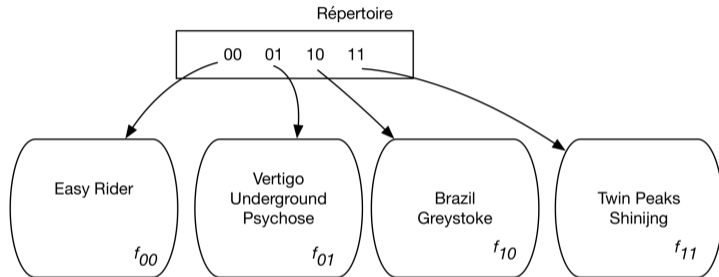
**Cas 1** : on insère dans un fragment plein, mais plusieurs entrées pointent dessus  
⇒ on alloue un nouveau fragment, et on répartit les pointeurs

**Cas 2** : on insère dans un fragment plein, associé à une seule entrée  
⇒ on double à nouveau le nombre d'entrées

Simple, mais la croissance du répertoire est un potentiel problème.

# Greystoke 10111001 et Shining 11010011

Le fragment 1 déborde. Il était référencé par deux entrées du répertoire.



On ajoute un fragment, on redistribue, le répertoire ne change pas. C'est le cas qui devient le plus courant quand la collection grandit.



# À retenir : hachage extensible

Le hachage extensible résout en partie le principal défaut du hachage, l'absence de dynamicité.

Le répertoire tend à croître de manière **exponentielle**, ce qui peut soulever un problème à terme.

Il reste une structure **plaçante** qui doit être complétée par l'arbre B pour des index secondaires.