

Bases de Données & Optimisation : Index Haché...

Thomas Gerald

September 30, 2025

Laboratoire Interdisciplinaire des Sciences du Numérique – LISN, CNRS
`thomas.gerald@lisn.upsaclay.fr`

Index et Hachage

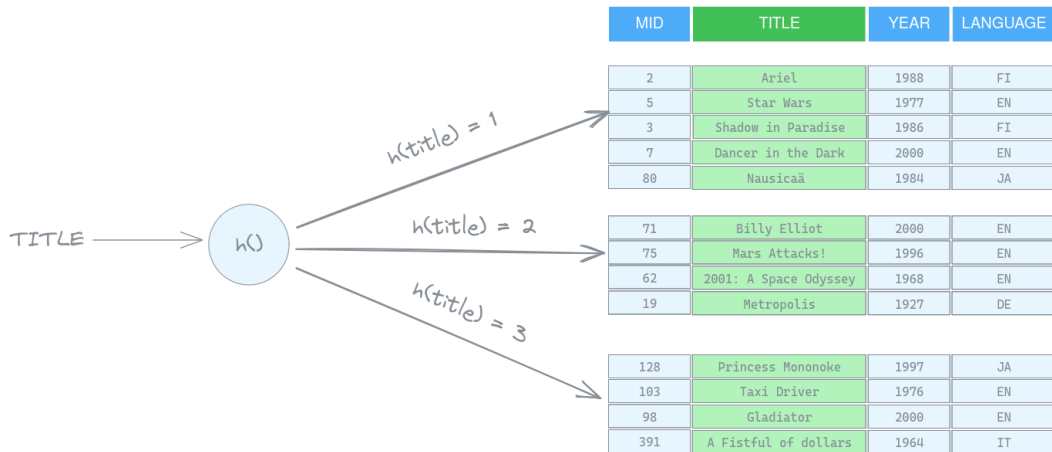


Figure 1: Un fichier Haché

Hachage statique :

- Le nombre de “buckets” est fixé (groupe d’entrées de l’index)
- Ajout d’élément dans un bucket plein (“overflow”):

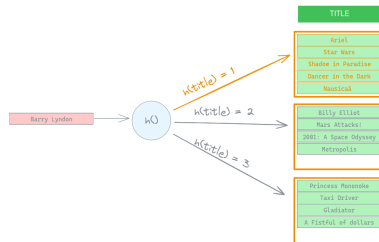


Figure 2: Index Haché statique

Hachage statique :

- Le nombre de “bucket” est fixé (groupe d’entrées de l’index)
- Ajout d’élément dans un bucket plein (“overflow”):
 - Création d’une nouvelle page dans le “bucket” (avec la nouvelle entrée)

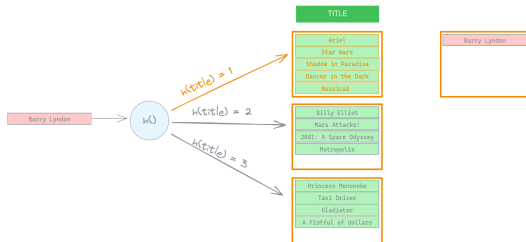


Figure 3: Index Haché statique

Hachage statique :

- Le nombre de “bucket” est fixé (groupe d’entrées de l’index)
- Ajout d’élément dans un bucket plein (“overflow”):
 - Création d’une nouvelle page dans le “bucket” (avec la nouvelle entrée)
 - Crée une référence dans la page précédente sur la nouvelle page

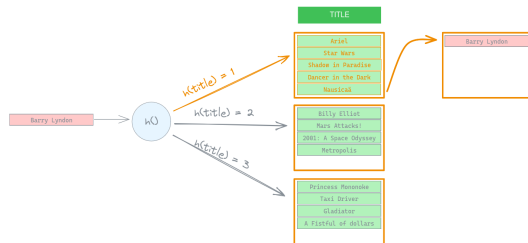


Figure 4: Index Haché statique

Index Haché statique

Hachage statique :

- Le nombre de “bucket” est fixé (groupe d’entrées de l’index)
- Ajout d’élément dans un bucket plein (“overflow”):
 - Création d’une nouvelle page dans le “bucket” (avec la nouvelle entrée)
 - Crée une référence dans la page précédente sur la nouvelle page

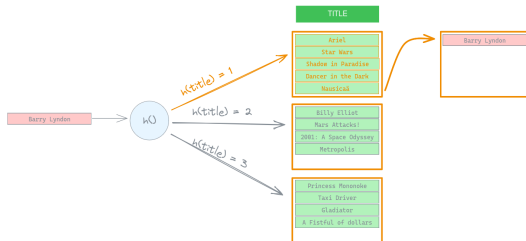


Figure 5: Index Haché statique

- Si pas assez de “bucket” → peut-être proche du tas en terme de recherche !!!
- Si trop de “bucket” → Beaucoup d’espace utilisé avec peu d’entrées (stockage) !!!

Index Haché : Quelques précisions

Index Haché

On va considérer $h : X \rightarrow \mathbb{N}$ la fonction de hachage

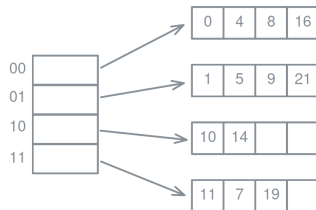
- On considère la représentation binaire de $h(x)$
- Pour retrouver le bucket on associe à une partie tronquer de $h(x)$



Index Haché dynamique

Hachage dynamique :

- Le nombre de “bucket” est dynamique
- Ajout d’élément dans un bucket plein (“overflow”):
 - Création d’un nouveau bucket (au moins)



Dupliquer l'index

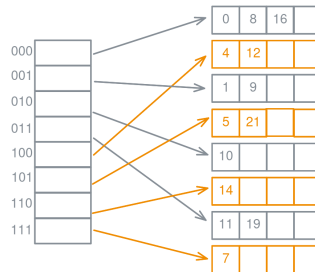
Index Haché dynamique

Hachage dynamique :

- Le nombre de “bucket” est dynamique
- Ajout d’élément dans un bucket plein (“overflow”):
 - Création d’un nouveau bucket (au moins)

Dupliquer l’index

- Passer des deux derniers de bits $\rightarrow$ 3 bits pour encoder les buckets



Index Haché dynamique

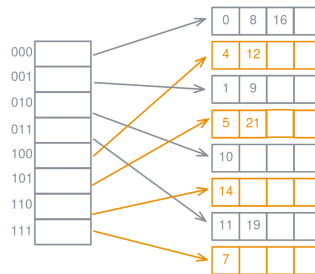
Hachage dynamique :

- Le nombre de “bucket” est dynamique
- Ajout d’élément dans un bucket plein (“overflow”):
 - Création d’un nouveau bucket (au moins)

Dupliquer l’index

- Passer des deux derniers de bits $\rightarrow$ 3 bits pour encoder les buckets
- **Dupliquer le nombre de pages** (ici de 4 à 8 pages)

Dupliquer le nombre de pages

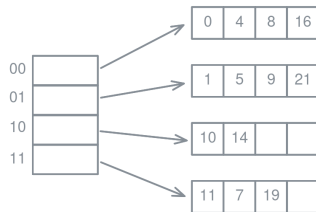


Index Haché dynamique

Hachage dynamique :

- Le nombre de “bucket” est dynamique
- Ajout d’élément dans un bucket plein (“overflow”):
 - Création d’un nouveau bucket (au moins)

Dupliqué le nombre de page



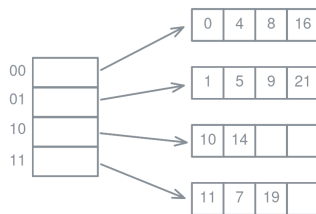
Index Haché dynamique

Hachage dynamique :

- Le nombre de “bucket” est dynamique
- Ajout d’élément dans un bucket plein (“overflow”):
 - Création d’un nouveau bucket (au moins)

Éviter la duplication des pages

Creation d’un nouveau bucket



Index Haché dynamique

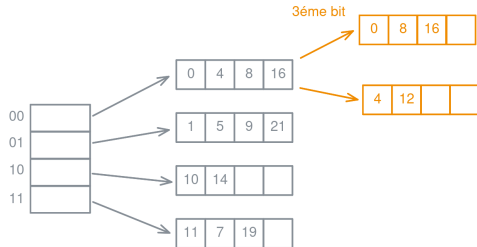
Hachage dynamique :

- Le nombre de “bucket” est dynamique
- Ajout d’élément dans un bucket plein (“overflow”):
 - Création d’un nouveau bucket (au moins)

Éviter la duplication des pages

- Séparer seulement le bucket plein

Création d’un nouveau bucket



Index Haché dynamique

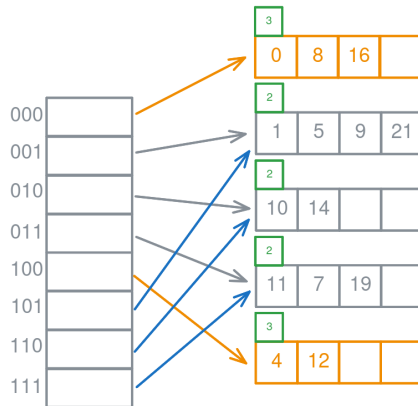
Hachage dynamique :

- Le nombre de “bucket” est dynamique
- Ajout d’élément dans un bucket plein (“overflow”):
 - Création d’un nouveau bucket (au moins)

Éviter la duplication des pages

- Séparer seulement le bucket plein
- Dupliquer la table pointant vers les buckets
- Garder dans les pages l’information sur le nombre de bits utilisés

Création d’un nouveau bucket



Les autres index

Les index inversés (recherche de texte) :
Faire des recherches sur les mots d'un texte:

- Des textes long comme des commentaires
- Des listes...

Quelques implémentations

- GIN Indexes dans postgres^a
- InnoDB Full-Text Indexes^b

^a<https://www.postgresql.org/docs/current/gin-intro.html>

^b<https://dev.mysql.com/doc/refman/8.0/en/innodb-fulltext-index.html>

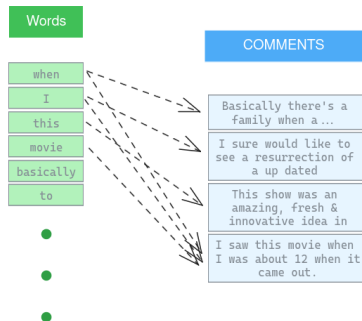


Figure 6: Exemple d'un index inversé

Les index bitmap :

Une table contenant des valeurs binaires où la valeur vaut 1 si la condition est respectée.

Utilisation quand on a :

- Des valeurs qui se répètent
- Peu de valeurs différentes

IS EN	LANGUAGE
0	DE
0	IT
1	EN
1	EN
1	EN
0	JA
0	FI
0	FI
1	EN
0	JA
1	EN
1	EN
1	EN

Objectifs :

- Être en mesure de dérouler les algorithmes d'insertion, de suppression et de recherche pour les index B-tree, Trié et Haché

Objectifs :

- Être en mesure de dérouler les algorithmes d'insertion, de suppression et de recherche pour les index B-tree, Trié et Haché
- Être capable de choisir l'index le plus adapté pour un type de requête particulier

Objectifs :

- Être en mesure de dérouler les algorithmes d'insertion, de suppression et de recherche pour les index B-tree, Trié et Haché
- Être capable de choisir l'index le plus adapté pour un type de requête particulier
- Mesurer le nombre de lecture et d'écriture de pages pour les différents algorithmes utilisés dans l'indexation

Objectifs :

- Être en mesure de dérouler les algorithmes d'insertion, de suppression et de recherche pour les index B-tree, Trié et Haché
- Être capable de choisir l'index le plus adapté pour un type de requête particulier
- Mesurer le nombre de lecture et d'écriture de pages pour les différents algorithmes utilisés dans l'indexation
- Être capable de calculer la complexité des différents algorithmes vus dans le cours

Objectifs :

- Être en mesure de dérouler les algorithmes d'insertion, de suppression et de recherche pour les index B-tree, Trié et Haché
- Être capable de choisir l'index le plus adapté pour un type de requête particulier
- Mesurer le nombre de lecture et d'écriture de pages pour les différents algorithmes utilisés dans l'indexation
- Être capable de calculer la complexité des différents algorithmes vus dans le cours
- Dans quel cas utiliser les index ?