

**Connolly et Begg - Chapitres 18**

**Indexation**

Marc Philippe Parent Database Systems 6e de Connolly et Begg

# Base de données

**IFT-2004**

**Département d’informatique et de génie logiciel**



# Objectifs

* Organisation fichier
* Type d'index
* Sélection d’index
* Suppression d’index
* Optimiseur
* Création d’index avec SQL.

Connolly et Begg Database systems 6th 2

Département d’informatique et de génie logiciel – Marc Philippe Parent



# Organisation de fichier

* Heap

C’est un tas, tout est placé désordonnée

Il y a pas d’ordre alphabetique, c’est l’ordre d’arrivée qui fait l’ordre.

Avantange : Insertion rapide. Car c’est à la suite.



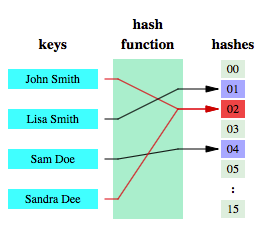
* Indexed Sequential Access Method (ISAM)

Fichier classé, si tu cherche un utilisateur par son

Login le SGBD va avoir moin de difficulté

Recherche Binaire : Si ta une liste classé

alphbétiquement LOG N /2



* Hash

fonction mathématique qui renvoi une valeur

* + Calcul hachage
  + Collision

Le hachage multiple est lié a à la collision

Adresse ouvert, case suivante 2 est déjà

On l’Envoi dans 3.

* + Hachage \_\_\_Dynamique\_\_\_\_\_\_

La fonction est adapté en fonctions du nombre

D’enregistrement, exemple au lieu d’avoir une fonction de hachage basé sur les 26 lettres de l’alphabet pour retrouver des mots dans le dictionnaire, on a une fonction de hachage qui est basé sur le nombre de mots a l’intérieur de ce dictionnaire.

* + Limité.

Connolly et Begg Database systems 6th 3

Département d’informatique et de génie logiciel – Marc Philippe Parent

# Types d'index

Si c’est classé sur la clé primaire ce n’est pas classé par le numéro de téléphone les indexes sont fait par le NO!

Il faudras alors faire d’autres indexes si on veux faire des recherches par d’autre champs.

* Primaire

indexer par la clé primaire.

* De cluster

Si la clé primaire c’est le NO,EMP mais tu classe par une clé alternative c’est un index de cluster.

* Secondaire

C’est tout les autres attributs, sa va classer des index pour prenom, nom courriel, on peut créer des index poru toutes les colonne si nous voulons.

* Qualité

Dense et clairsemé, quand un index est petit, si tes des millions et milliard d’enregistrement ton index deviendrais trop gros, Au lieu de contenir tous les elements il va garder une valeur proche.

Connolly et Begg Database systems 6th 4

Département d’informatique et de génie logiciel – Marc Philippe Parent

# Choisir un Index

**Approche 1: pas classé,**

**Approche 2: classé.**

* Approche 1

on prend un heap, puis après on crée autant d’index secondaire qu’on a besoin.

* + Garder tuples \_\_\_désordonnées\_\_\_\_\_ (heap)
  + Créer \*\_ index secondaires
* Approche 2
  + Ordonner tuples index primaire ou cluster
    - Attribut utilisé plus souvent jointure

Je fais beaucoup de jointure avec ma clé candidate je vais utiliser cela comme index de cluster.

Le plus souvent c’est classé par index primaire.

* + - Attribut utilisé plus souvent pour \_\_accéder\_\_\_\_\_ les tuples dans cet ordre.

Connolly et Begg Database systems 6th 5

Département d’informatique et de génie logiciel – Marc Philippe Parent

# Choisir un Index

* Si \_PK\_
  + Index primaire (HEAP + B-Tree) / IOT
* Sinon
  + Index de cluster
* \_\_1..1\_\_\_index primaire/de cluster par table
* Index \_\_\_\_secondaires\_\_\_\_\_\_\_
  + B-Tree / Bitmap / Fonction (FBI)

(FBI = Function Base Index) : C’est la valeur absolue de chaque valeur et crée un index à partir de cela.

Arbre binaire.

* + Mécanisme pour spécifier \_clé\_\_ additionnelle

[18:10]: toute les AK quand je les ai crée en disant qu’Elle etaient unqiue cela crée des index secondaire unique.

* + Retrouver données + efficacement
  + Surcharge de \_\_\_Maintenance\_\_\_\_\_\_\_\_.

Parce que le SGBD doit maintenir les index, a chaque fois qu’il y a de la mise à jour, l’index doit être mis à jour, si il y a un ajout de donnée, l’index doit être mis à jour. Il faut faire attention un moment donné sa va couter cher en maintenance.

Connolly et Begg Database systems 6th 6

Département d’informatique et de génie logiciel – Marc Philippe Parent

# Maintenance d’index

* Tuple ajouté
  + Ajouté enregistrement index secondaires
  + Tuple MAJ.
    - * MAJ index secondaire
  + Augmenter espace Disque
  + Possible dégradation de performance

Si il y a trop d’index le SGBD va perdre trop de temps a évaluer le chemin le plus rapide au lieu d’éxécuter la requête.

Connolly et Begg Database systems 6th 7

Département d’informatique et de génie logiciel – Marc Philippe Parent

# Principaux types d’index

* B+-Tree

B Tree qui represente la division par 2 des bottins.

* + Table Organisée en Index Index Organized Tables (IOT)

Index organised table,

* + Regroupement Cluster

physiquement, tu a les clients et les facture tu a 2 table, departement et employé

Le regroupement de cluster va classer les employé et les départements ensemble.

On le fait souvent parce qu’il a souvent des jointures.

* Bitmap
  + (plus loin)
* Index Fonctionnels - Function Based Index (FBI)
  + Stockés dans \_B+-Tree\_\_\_
  + Stockés dans \_\_Bitmap\_\_\_\_

Si vous faite souvent le fonction upper(NOM) il peut être bon de faire un index sur les valeurs calculé de upper car miniscule et majuscule sont des valeurs différentes.

Connolly et Begg Database systems 6th 8

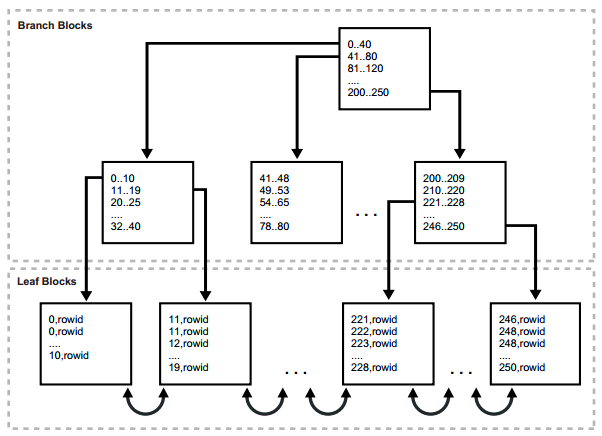
Département d’informatique et de génie logiciel – Marc Philippe Parent

# Principaux types d’index – B+-Tree

# Il va comparer le nombre avec les blocs. Si je veux retrouver

# L’employé 223 il va chercher

# le rowid est le numéro physique de l’enregistrement sur le disque (un pointeur vers une adresse mémoire)

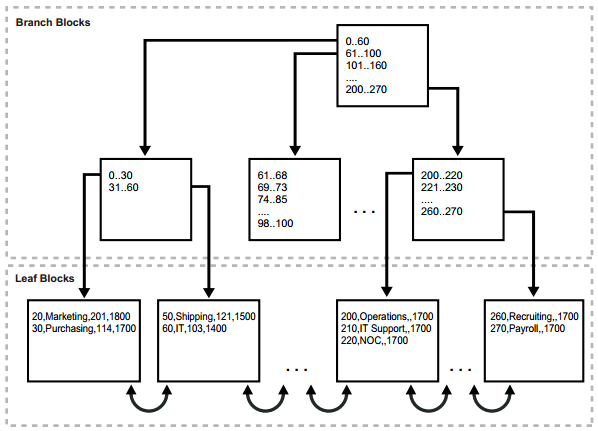


Connolly et Begg Database systems 6th 9

Département d’informatique et de génie logiciel – Marc Philippe Parent

# Principaux types d’index – IOT

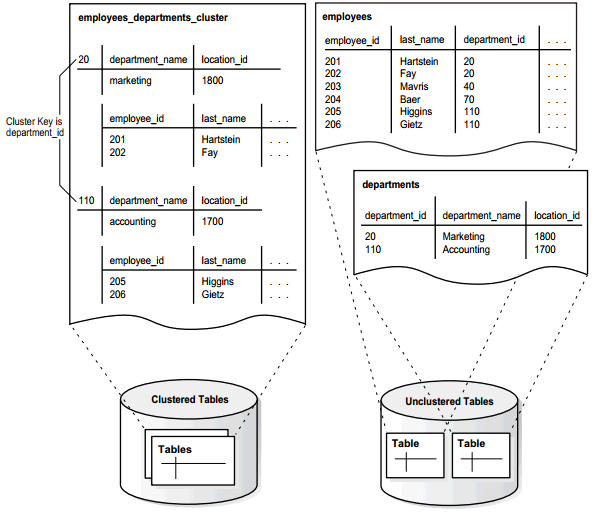
# La différence c’est que l’enregistrement est directement dans l’index.



Connolly et Begg Database systems 6th 10

Département d’informatique et de génie logiciel – Marc Philippe Parent

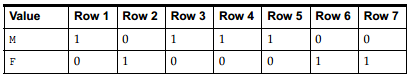
# Principaux types d’index – Cluster



Connolly et Begg Database systems 6th 11

Département d’informatique et de génie logiciel – Marc Philippe Parent

# Principaux types d’index – Bitmap



Connolly et Begg Database systems 6th 12

Département d’informatique et de génie logiciel – Marc Philippe Parent

# Choisir Index secondaires

1. Pas \_\_\_petites\_\_\_\_ relations (critère négatif)

Ne pas faire des indexes sur des petites tables. (on ne prevois pas bcp de tuples dans le temps.)

1. PK si pas \_clé\_\_ de l'organisation de fichier
2. \_FK\_ si fréquemment accédée

Si elle est souvent utilisée c’est cohrent.

1. Tout attribut beaucoup utilisé comme \_clé\_\_ recherche secondaire

si c’est un champ souvent dans un where.

1. Attributs impliqués dans (\_select\_\_\_\_, critère jointure, order by, group by et Autre opérations de \_tri\_\_ (union ou distinct)

6. \_\_Fonction\_\_\_\_ système

meme faire un index de fonction.

1. \_\_Plan\_\_ index-seulement

Si fais une requete sur le numéro téléphone et tu fais juste afficher le téléphone, à ce moment c’est une bonne idée de faire un index sur ce champ.

1. Éviter attribut/relation \_Maj\_\_ fréquemment (critère négatif)

éviter de mettre des index sur des attributs souvent mis à jour. Si les champs sont modifié (insert, delete) plus souvent qu’ils sont selectionné il est mauvais d’indexer.

1. Éviter si requête retourne \_\_majorité\_\_\_\_ de tuples (critère négatif)
2. Éviter longue \_\_chaine\_\_\_\_ de caractères. (critère négatif)

Si les champs sont trop grand mettez pas d’index. Ex varchar2  2000 c’est trop gros.

Connolly et Begg Database systems 6th 13

Département d’informatique et de génie logiciel – Marc Philippe Parent

# Exemple

# A cause du or il faudrais améliorer les 3 critères.

* Attributs TYPE et LOCATION \_\_indexés\_\_\_\_\_ dans PROPRIETE\_A\_LOUER

select \* from PROPRIETE\_A\_LOUER where (TYPE = 'Appartement'

or LOCATION > 500 or PIECES > 5) ;

* 2 index servent pour 2 conditions mais pas \_\_pièces\_\_\_\_<5
* Si critère plus d’un \_\_prédicat\_\_\_\_\_\_
  + Si un des termes contient or et terme n’impose pas d’ordre d’index ou de tri
    - \_Pas\_\_ créer index
* Si prédicats de where avaient été assemblés avec des \_and\_\_
  + 2 index \_\_exploités\_\_\_\_\_\_\_.

Connolly et Begg Database systems 6th 14

Département d’informatique et de génie logiciel – Marc Philippe Paren

# Supprimer les index de la liste souhaités

* Mesurer \_impact\_\_ sur MAJ
* Maintenance  forts \_\_\_\_ralentissements\_\_\_ MAJ
* Index peut rendre plus \_\_\_efficace\_\_\_\_\_ MAJ

il arrive de mettre à jour un numéro de téléphone par un champ pas indexé

* Expérimenter

Déterminer si index améliore un peu ou réduit

Si vous mettez un index par prenom en premier et nom ensuite, et que les gens recherche par nom, et ensuite prenom, ce serait beaucoup de refaire l’index par nom et ensuite prenom.

* Réduction majeure de performances
  + Supprimer index
* Faible amélioration.
  + Vérifier circonstances

Connolly et Begg Database systems 6th 15

Département d’informatique et de génie logiciel – Marc Philippe Parent

# Optimiseur

* SGBD permet inspection stratégie optimiseur
  + \_\_Plan\_\_ d’exécution de requête
* Oracle utilitaire de diagnostic \_\_\_\_EXPLAIN.PLAN\_\_\_\_\_\_\_\_

Dans oracle, il y a un onglet qui sappel plan d’execution

Tu envoi une requete pour voir comment il a executer la requete : quel index il a prit.

* Requête plus lente que prévu
  + Peut déterminer raison
  + Peut trouver \_\_\_stratégie\_\_\_\_\_\_ alternative
  + Exemple
* \_\_Insérer\_\_\_\_\_ grand nombre tuples
  + Si insertions augmenter taille + de \_10%\_\_\_:
    - Supprimer \_\_\_\_temporairement\_\_\_\_ index.

Peut être plus rapide que de modifier l’index.

Connolly et Begg Database systems 6th 16

Département d’informatique et de génie logiciel – Marc Philippe Parent

# Statistiques et documentation

* Optimiseur de requête repose sur statistiques BD
  + Contenues dans catalogue
* Peut nécessiter \_\_\_utilitaire\_\_\_\_\_\_\_ pour MAJ catalogue
* Choix index soigneusement documenté.
  + Surtout -> quand pas indexer attributs

Documenter pourquoi avoir crée tel index pour quel requetes etc. parce que

Pour améliorer tel requete pour tel requete

On a pas fait cette requete parce que sa a ralentit tel chose.

Pour ne pas repeter les erreurs et éviter la perte de temps!

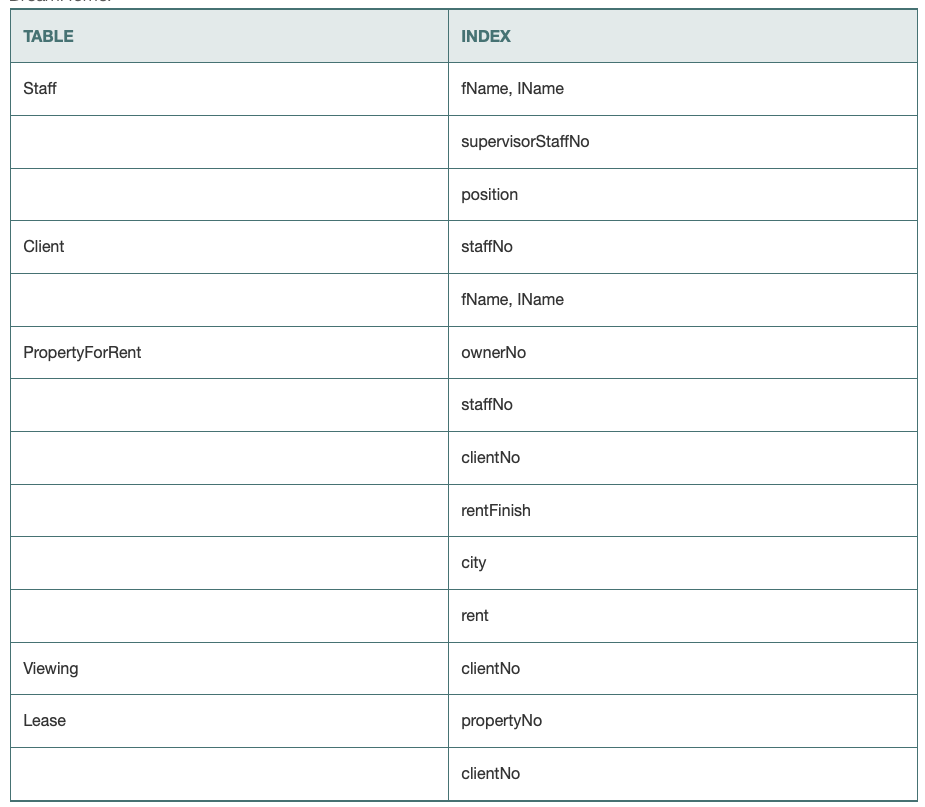
Connolly et Begg Database systems 6th 17

Département d’informatique et de génie logiciel – Marc Philippe Parent

# Maison de rêve

# Remarquer ici il n’y a aucune clé primaire

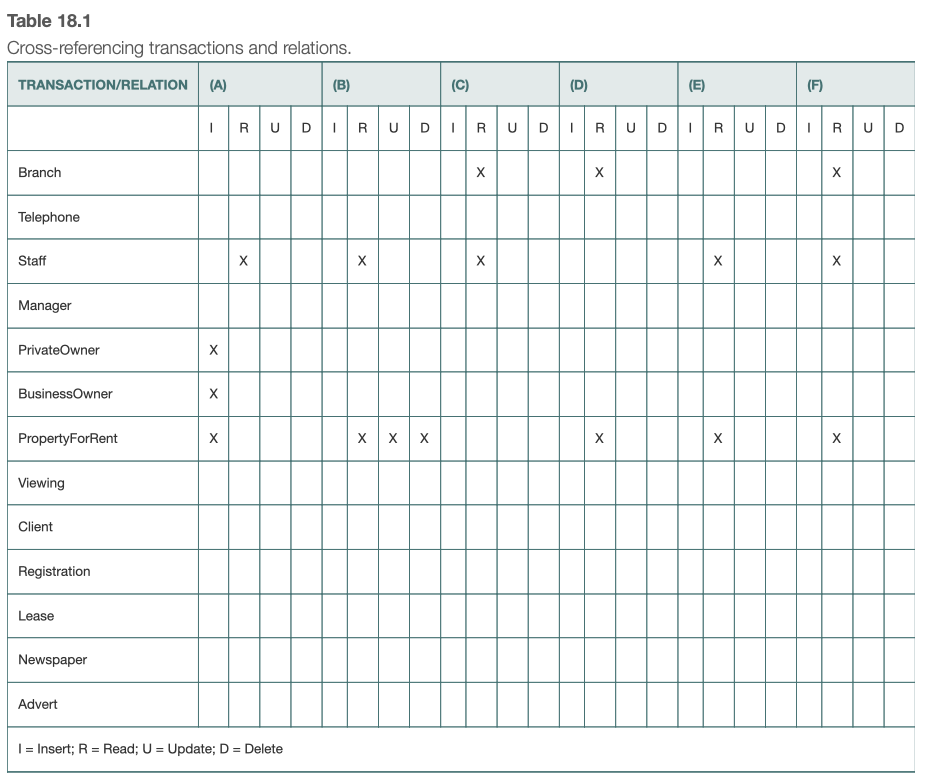
# Elles sont déjà indexés



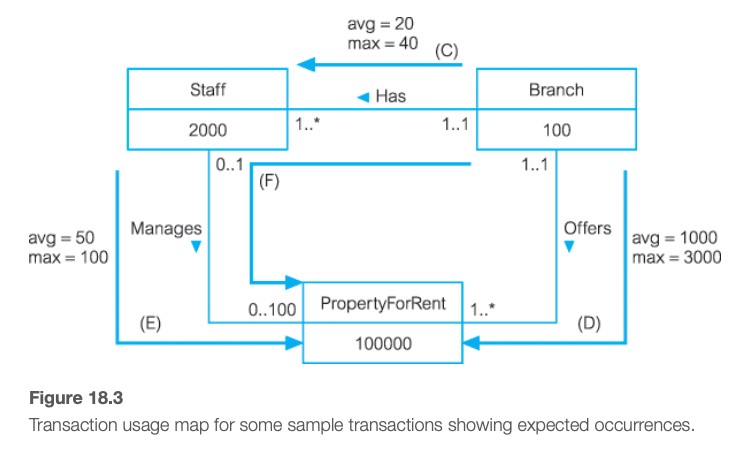
Connolly et Begg Database systems 6th 18

Département d’informatique et de génie logiciel – Marc Philippe Parent

# Maison de rêve



*•*



Connolly et Begg Database systems 6th 19

Département d’informatique et de génie logiciel – Marc Philippe Parent

# Créer un index secondaire

* Syntaxe

create [unique] index NOM\_INDEX

on NOM\_TABLE(COLONNE1,… COLONNEn);

* Notes
* Exemples.

Connolly et Begg Database systems 6th 20

Département d’informatique et de génie logiciel – Marc Philippe Parent



# Exercices

* Lequel n'est pas un bon critère pour attribuer un index secondaire?

A-Clé étrangère fréquemment accédée B-Attribut utilisé dans un order by

C-Attribut utilisé dans un group by

D-Attribut utilisé dans une fonction système E-Aucune de ces réponses

* Lequel est un bon critère pour ne pas supprimer un index secondaire?

A-Petite relation

B-Plan index-seulement

C-Attribut/relation MAJ fréquemment D-Requête retourne majorité de tuples E-Longue chaîne de caractères.

Connolly et Begg Database systems 6th 21

Département d’informatique et de génie logiciel – Marc Philippe Parent