

**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

* Organisation Heap

C’est de l’information désordonnée c’est l’ordre d’arrivée qui fait l’ordre.

L’avantage du heap est que les insertions sont rapide.

Le désavantage est que les recherches sont lentes.

* Indexed Sequential Access Method (ISAM)

C’est une index qui classe les informations en ordre selon une certaine valeur

Ce peut être la valeur de la clé primaire ou autres attributs ou méthodes

L’avantage est que les recherches sont plus rapide.

Par contre l’insertion moin rapide.

L’index est utilisé à l’aide d’une recherche Binaire (b-tree) :

Celle-ci sépare chaque comparaisons en 2 blocs avec le même nombre de données.

Organisation Hash

* + Calcul hachage

C’est une fonction mathématique qui renvoit une valeur (une position),

Cette valeur correspond à l’endroit ou l’enregistrement est insérée.

* + Une collision survient lorsque :

2 ou plusieurs enregistrements se retrouve au même endroit lors d’un hachage.

Hachage multiple :

c’est une 2e fonction de hachage qui s’execute lorsque qu’une collision survient.

* + Hachage Dynamique

C’est une fonction de hachage qui :

s’adapte à la quantité de valeurs : Elle va s’auto-modifier pour s’adapter.

Le hachage est Limité car il :

ne peut pas se faire sur nimporte quel champ (il doit être unique)

Connolly et Begg Database systems 6th 3

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

# Types d'index

* Index Primaire

C’est un index qui est fait sur le clé primaire.

* De cluster

C’est l’opposée de l’index primaire, celle-ci permet de classer la table avec une autre clé que la clé primaire.

Attention l’ordre de la table peut être

soit primaire ou cluster il ne peut pas être les deux à la fois.

* Secondaire

Ce sont les index fait sur les autres attributs et clés ou regroupement d’attribut / clés

* Il y a deux qualités d’index :

Clairsemé: il va prendre seulement les valeurs principales pour ne pas avoir un index trop grand. (seulement une partie des valeurs se retrouveront dans l’index)

Dense: Toutes les données sont a l’intérieur.

Connolly et Begg Database systems 6th 4

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

# Choisir un Index

L’Approche 1 consiste à

* + Garder les tuples de façon désordonnée (heap)
  + Et Créer autant d’index secondaires que nécessaire
* L’Approche 2 consiste à
  + Ordonner les tuples avec un index primaire ou de cluster
    - Et indexer les attributs / clé utilisés souvent en jointure
    - Ainsi que les attributs utilisé plus souvent pour accéder aux tuples dans cet ordre.

Connolly et Begg Database systems 6th 5

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

# Choisir un Index

* Lorsque l’index est fait sur la PK :
  + Index primaire fonctionne de facon

HEAP + B-Tree / IOT (Index Order Table)

Ce qui veut dire: Insertion par Heap et recherche par B-Tree.

* Sinon
  + Index de cluster
* Il est seulement possible de mettre 1 index primaire ou de cluster par table et pas les deux.
* Index secondaires
  + B-Tree / Bitmap / Fonction (FBI) Function Base Index

Bitmap index sont des index fait avec des booléens

Function Base Index : sont des index avec des fonctions appliqué sur le champ

L’index secondaire c’est aussi :un mécanisme

pour spécifier des clés additionnelle par exemple un AK.

* + Les index secondaire permet de retrouver données + efficacement
  + Mais elle provoque :

Une surcharge de Maintenance.

Parce que le SGBD doit maintenir l’index à chaque insertion.

Connolly et Begg Database systems 6th 6

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

# Maintenance d’index

* Dès qu’un tuple est
  + Ajouté
  + MAJ.
    - * Il y aura MAJ de l’index secondaire
  + Cela va 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 recherche.

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

|9:6| : Si 2 table sont souvent jointes, les 2 tables vont être stockés ensemble.

physiquement, tu a les clients et les facture tu a 2 table, département 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.

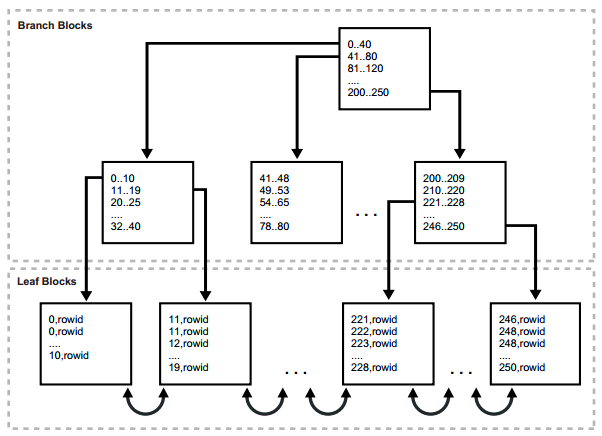
Connolly et Begg Database systems 6th 8

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

# Qu’elle type d’index represente ce graphique?

# Ps : ROWID

# – B+-Tree

****

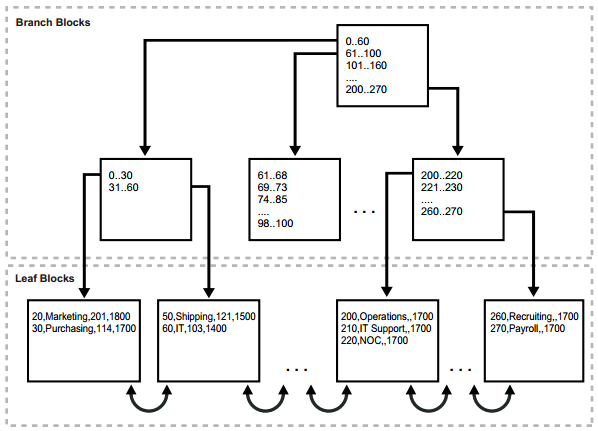
Connolly et Begg Database systems 6th 9

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

# Qu’elle type d’index represente ce graphique?

# PS : ENGISTREMENT

# – IOT (Index Order Table)

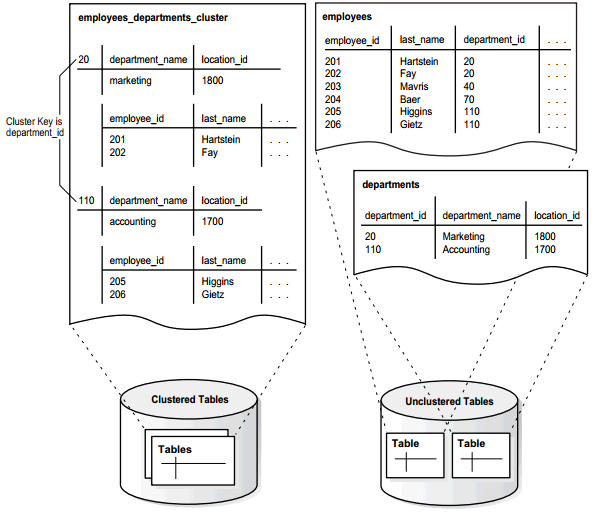
****

Connolly et Begg Database systems 6th 10

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

# Principaux types d’index – Cluster

# |9:11| : Classer les 2 tables ensemble. Ex no de département avec ses employé en dessous.

****

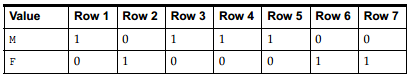
Connolly et Begg Database systems 6th 11

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

# Principaux types d’index – Bitmap

# |9:13| : Bitmap pour valeur 0 ou 1.

# |9:12| : Souvent utilisé dans les engins OLAP sera vu vers la fin du cours.

****

Connolly et Begg Database systems 6th 12

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

# Choisir Index secondaires

**Qu’elles sont les bonnes raisons de choisir un index secondaire?**

1. choisir un index sur la clé primaire si l’organisation de la table à été fait par Cluster.
2. Choisir un index sur clé étrangère si fréquemment accédée
3. Tout attribut beaucoup utilisé comme clé recherche secondaire (where)
4. Attributs impliqués dans

(un select, critère jointure, order by, group by et Autre opérations de tri (union ou distinct)

1. Fonction système

(lorsque vous utilisez souvent des fonctions sur des attributs)

1. Plan index-seulement:

Une requête qui utilise seulement l’index sans fouiller dans la table principale.

# Choisir Index secondaires

**Qu’elles sont les raisons pour ne pas choisir de faire un index secondaire?**

1. Éviter l’index si les requêtes retourne la majorité des tuples
2. Éviter l’index sur des longues chaines de caractères.
3. Éviter l’index sur des attributs ou relation

mis à jour (insert/update/delete) fréquemment

1. Éviter l’index sur des petites relations

Connolly et Begg Database systems 6th 13

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

# Exemple

* 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 le critère contient plus d’un prédicat
  + Et si un des termes contient or et le terme n’impose pas d’ordre d’index ou de tri
    - Ne 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

|9:20| : Vous changer le salaire d’une personne en recherchant la personne par le NAS.

|9:21| : si le NAS n’est pas indexé sa serait bien de le faire.

* 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 mieux 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 s’appel 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.
* |9:41| : Il y a une commande pour mettre a jour les statistique. Dans oracle
  + 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!

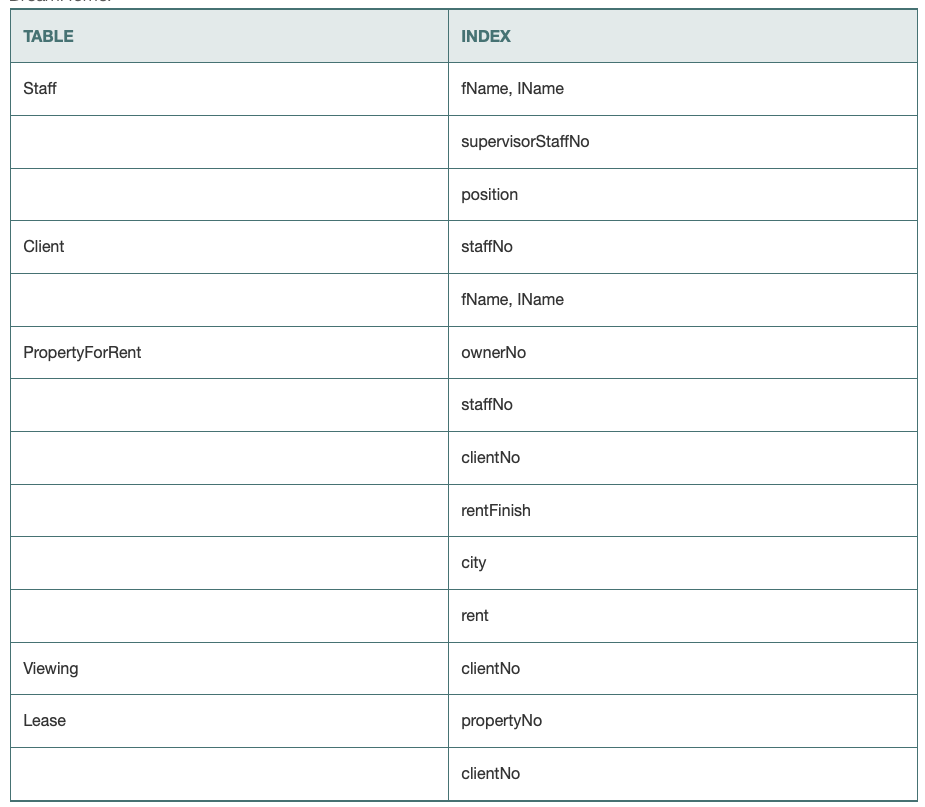
Connolly et Begg Database systems 6th 17

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

****

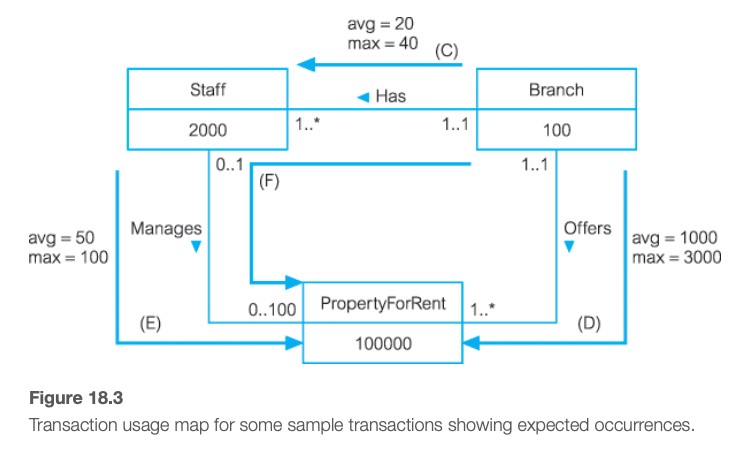
Connolly et Begg Database systems 6th 18

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

|9:43| : Tout les index ont été fait sur des clés étrangères ou autres champs dans ce cas.

# Maison de rêve

*•*

****

**|9:44| : Montre les type d’operations qui sont fait souvent pour chacune des tables.**

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