

Système de Base de données Relationnelles

Optimisation de requêtes

M1 Informatique – Développement Logiciel
Semestre 7

Cours donné par Abdelkader HAMEURLAIN
Rédigé par Antoine de ROQUEMAUREL

2014

Avant-propos

L'objectif de l'optimisation est de minimiser le temps de recherche. Pour cela, on définit une stratégie ou méthode d'optimisation selon 3 critères :

- Plan d'exécution
- Espace de recherche
- Modèle de coûts

De ces 3 critères découlent l'obtention du Plan d'exécution le plus proche de l'optimal. (cf. schéma)



Le plan d'exécution n'est jamais le plus optimal possible car il engendre d'énormes coûts pour des différences moindres.

Table des matières

1	Les opérateurs physiques	4
1.1	Système de gestion de fichier	4
1.2	Algorithme de jointure	4

1

Les opérateurs physiques

1.1 Système de gestion de fichier

Definition 1.1 **Fichier** collection d'article (de fichiers) contenant de l'information (information de gestion).

Definition 1.2 SGF ^a programme permettant à l'utilisateur de :

- Définir la structure d'un fichier
- Créer un fichier
- Interroger un fichier
- Modifier un fichier

a. Système de Gestion de Fichier

1.2 Algorithme de jointure

1.2.1 Jointure par produits cartésiens

Principe : Comparaison de tous les tuples de la relation R avec tous les tuples de la relation S.

1.2.1.1 Jointure par produit cartésien sans index

$$\text{Nombre de pages lues} = |R| \times |S| + |R|$$



Pour des raisons de performance, on privilégiera la plus petite relation comme relation externe.

Relation externe = R

Inconvénients

- Nombre important d'entrées/sorties disque
- Pour chaque tuple de R la totalité de S est balayé

1.2.1.2 Jointure du produit cartésien par ind

- La création d'un index permet d'éviter que, pour chaque tuple R, la totalité de S soit balayé.

- L'utilisation d'un index se fait sur l'attribut de jointure de la relation interne (S) sous la forme (attribut, pointeur).

On obtient alors :

Index plein

$$nb_{pages} = \frac{nbTuples \times (taille(attribut) + taille(pointeur))}{taille(page)}$$

Index creux

$$nb_{pages} = \frac{nbPagesPrecedents \times (taille(attribut) + taille(pointeur))}{taille(page)}$$

Tant que l'on obtient pas un nombre de page (nbPages) égal à 1 on continue.



On arrondi ici toujours le nombre le nombre de page à la valeur entière supérieure (on ne lit pas une moitié de page).

$$tempsLecture = (nb_{index} + 1) \times tempsAccesDisque$$

ou le nombre d'index

$$nb_{index} = 1 \times index_{plein} + n \times index_{creux}$$