

Les index

M1 Statistique

E. Claeys

ICUBE/IRMA
Université de Strasbourg

Base de données, 2018

Programme du cours

- 1 Pourquoi un index
- 2 Qu'est ce qu'un index
- 3 La mémoire et les index

- Quelles sont les bonnes pratiques pour économiser la mémoire dans un SGBDD ?
- Quelles sont les contraintes que vous avez déjà vus sur des attributs
- Existe-il un ordre hiérarchique dans les tuples d'une tables ?

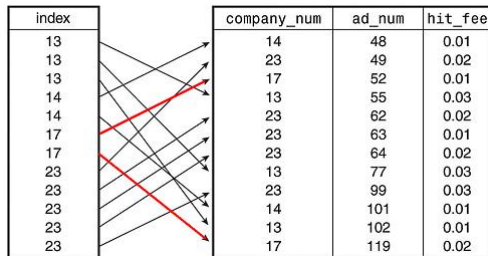


- Quelles sont les bonnes pratiques pour économiser la mémoire dans un SGBD ?
- Quelles sont les contraintes que vous avez déjà vus sur des attributs
- Existe-il un ordre hiérarchique dans les tuples d'une tables ?

- Définir un type d'attribut en adéquation avec la taille de ses modalités (exemple `SMALL INT`)
- `UNIQUE`, `NOT NULL`, ...
- Non

Les index

Les index permettent de créer un ordre hiérarchique entre les tuples (mais pas seulement). Les tuples sont donc **triés** ce qui permet d'**accélérer** les requêtes. Lorsque vous créez un index sur un attribut (par exemple l'*id*), MySQL stock cet index sous forme d'une liste d'adresse (l'adresse mémoire des tuples) respectant l'ordre hiérarchique que vous souhaitez (par exemple par *id* croissant).



Il est possible de définir des index sur n'importe quel type d'attribut. Vous pouvez également coupler plusieurs attributs (colonnes) sur un index.

Il existe 4 types d'index qui rajoute parfois une contrainte supplémentaire à l'attribut

- L'index de type "simple" c'est à dire INDEX
- L'index de type UNIQUE qui n'accepte aucun doublon.
- L'index de type FULLTEXT défini sur une colonne de texte, et qui permet de faire des recherches sur des chaînes de caractères (par exemple trouver "petitjean" en cherchant "petit")
- L'index de type SPATIAL qui définit une distance sur des séries temporelles ou des objets géométriques

Par exemple pour chercher un Marcel Dupont dans une table d'individus, il faut lire toutes les lignes, car il peut y en avoir plusieurs et par malchance l'un d'entre eux peut figurer en dernière position dans la table. Créer un index sur Nom + Prénom a donc son utilité. Pour rechercher un Marcel Dupont dans un index sur Nom + Prénom, on peut agir par dichotomie et donc éliminer un nombre important de cas, d'autant plus important que le nombre de données est important (échelle exponentielle).

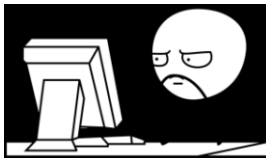
Les index : gain mémoire

L'efficacité d'une telle recherche est en $O(\log(n))$, c'est à dire que le nombre de cas scrutés et le gain entre lecture séquentielle des lignes et lecture dichotomique est le suivant :

Nombre de ligne	Nombre d'itération	Gain en %
5	3	40
10	4	60
50	6	88
100	7	93
500	9	98,2
1 000	10	99
5 000	13	99,74
10 000	14	99,86
1 000 000	20	99,998
5 000 000	23	99,99954
10 000 000	24	99,99976

Les index sont gourmands en mémoire ! Ne gaspillez pas votre capital « mémoire » en définissant un index sur un INT pour stocker par exemple la vitesse légale sur autoroute en France (par temps sec) : 130 km/h... Un TINY INT UNSIGNED suffira (permet de stocker les valeurs de 0 à 255). A quoi bon utiliser un INT dans cet exemple sachant qu'il occupe quatre fois plus de place qu'un TINY INT....

TD n° 5 Index en MySQL avec MySQL (30/50 minutes)



Pensez à sauvgarder vos commandes !