2.1.6 Indexation

Pour une gestion efficace des données, il est nécessaire de pouvoir identifier chaque tuple de façon unique.

L'indexation des données repose sur un principe simple d'étiquetage consistant à attribuer une étiquette différente à chaque jeu de données. Cette étiquette peut être une suite de caractères arbitraires, un entier, ou un descripteur explicite. On parle en général de clé ou d'identifiant pour désigner cette étiquette.

On peut représenter l'opération d'indexation sous la forme d'une fonction. Si t est le jeu de données, $k(t)$ désigne l'identifiant de ce jeu de données.



- L'**indexation** des données consiste donc à attribuer à chaque tuple distinct un **identifiant** unique.
- On parle également de *clé* du tuple, telle que:
  - soient t_1 et t_2 deux tuples,
  - si $k(t_1) = k(t_2)$, alors $t_1=t_2$.



L'exemple le plus simple d'indexation est celui fourni par les **numéros de case** d'un tableau.

- Soit T un tableau de n lignes
- le numéro $i < n$ est l'*index* de la ligne $T[i]$

Index	Données			
0	Dubois	Martine	29, rue du Verger, Orléans	22
1	Gilbert	Jonas	8, rue des Fleurs, Blois	23
2	Dalban	Pierre	13, av. du Général, Privas	22
	...	...	...	...
n - 1	Manoukian	Marianne	55, place des Bleuets, Aubagne	24

n



Autres exemples d'identifiants :

- numéro INE (étudiants)



- numéro URSSAF (sécurité sociale)
- numéro d'immatriculation (véhicules)
- numéro de compte en banque
- code barre
- etc.

Unicité

Il est nécessaire en pratique de définir une méthode permettant d'assurer l'unicité de la clé lors de l'ajout ou de nouvelles données dans le tableau de données. Il existe différentes méthodes permettant d'assurer l'unicité de la clé:

- Le gestionnaire de B.D maintient une liste triée des clés déjà utilisées. Lors de l'ajout d'un nouvel élément, il s'assure que la clé n'est pas déjà présente dans la liste.
- Dans le cas de clés à valeurs sur les entiers, il est possible d'utiliser un compteur dont la valeur sert de nouvelle clé et qui est incrémenté à chaque nouvelle insertion.
- Utilisation d'une *fonction de hachage* qui "calcule" la valeur de la clé à partir des valeurs contenues dans le jeu de données. La fonction de hachage doit être telle que la probabilité de choisir la même clé pour deux jeux de données différents soit extrêmement faible.

Efficacité

L'existence d'une clé unique pour chaque jeu de données permet la mise en œuvre d'une *recherche par clé* dans le tableau de données.

La recherche par clé repose sur l'existence d'une fonction d'adressage I qui à toute clé k associe la position i du tuple correspondant dans le tableau de données: $I : k \rightarrow i$. Ainsi si k est la clé servant à la recherche, l'extraction du tuple se fait en 2 étapes:

- $i = I(k)$ (recherche de l'adresse)
- $t = T[i]$ (lecture du tuple)

La recherche repose sur le parcours d'une structure de données ordonnée contenant les couples $(k_1,i_1), (k_2, i_2), ..., (k_N, i_N)$. Cette structure de données doit être organisée de telle sorte que la recherche soit la plus efficace possible. Dans le cas le plus simple, il s'agit d'une liste triée sur les clés $L = ((k_1,i_1), (k_2, i_2), ..., (k_N, i_N))$ telle que $k_1 < k_2 < ... < k_N$, de telle sorte que la recherche s'effectue en O(log N) (recherche dichotomique).

La clé doit en pratique tenir sur un nombre d'octets le plus petit possible pour que la liste L puisse tenir en mémoire. Autrement dit, il faut que :

- $|k| << |t|$

pour que :

- $|L| << |T|$



Une clé entière, codée sur 4 octets, permet d'indexer jusqu'à $2^{4 \times 8} \simeq 4\times10^9$ tuples différents

Previous : 2.1.5 Fichiers et Répertoires Up Next : 2.2 Aspect logique

From:
https://wiki.centrale-med.fr/informatique/ - **WiKi informatique**

Permanent link:
https://wiki.centrale-med.fr/informatique/public:std-3:cm1:aspect_physique:2.1.6_indexation

Last update: **2018/01/22 11:19**