III Sécurité

Protection des données contre les accès non autorisés

Contrôle d'accès ou autorisation

Les contrôles d'accès vérifient l'identité des usagers qui se présentent et en conséquence leur assignent des droits d'accès sur tel ou tel ensemble de données.

Autorisation (*GRANT* en SQL)

Tout usager qui a le droit de transmettre des privilèges sur un objet peut utiliser la commande *GRANT* pour transmettre ce privilège :

GRANT privilèges ON objet TO liste d'usagers
[WITH GRANT OPTION]

- Les privilèges peuvent être :
  - lire (*SELECT*),
  - insérer de nouveaux n-uplets (*INSERT*),
  - modifier des valeurs (*UPDATE*),
  - supprimer la totalité d'une relation (*DROP*),
  - créer de nouvelles relations (*CREATE*).

- L'option facultative *WITH GRANT OPTION* permet au donneur d'autoriser le receveur à transmettre à d'autres les privilèges qu'il reçoit.

- Un usager peut recevoir un privilège de plusieurs sources différentes.

Révocation (*REVOKE* en SQL)

Tout usager ayant donné un privilège peut à tout moment retirer ce privilège grâce à la commande *REVOKE* :

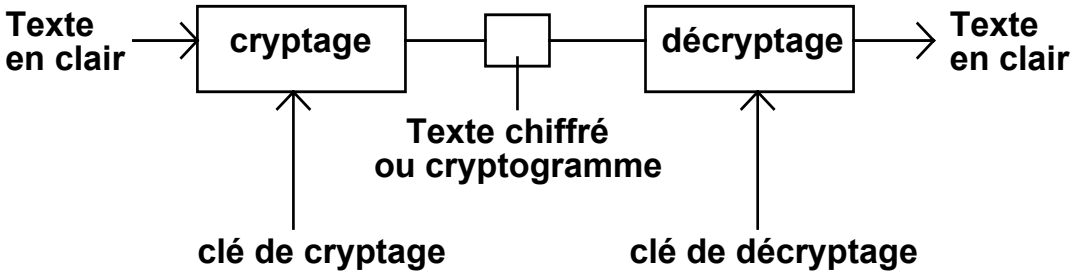
REVOKE privilèges ON objet FROM liste d'usagers

- Les privilèges sur l'objet mentionné sont retirés au receveur à moins que ce dernier ne les ait reçus d'une autre source, indépendante.

- Cette procédure de révocation complique le mécanisme d'autorisation car il faut appliquer recursivement les procédures de révocation puisqu'un usager auquel on retire un privilège a pu le transmettre à d'autres.

Cryptographie

La cryptographie a pour but de stocker ou de transporter l'information sous une forme telle que seuls les usagers en possession de la clé de décryptage sont susceptibles de la comprendre.



Cryptographie à clé publique

Elle fait appel à 2 clés

- une clé privée (gardée secrète par son détenteur) qui ne sert qu'au décryptage
- une clé publique qui n'est utilisée que pour crypter

L'algorithme de cryptage C et l'algorithme de décryptage sont choisis de telle sorte que le calcul de D soit très complexe même si l'on connaît complètement C

Ex.: Paul souhaite envoyer le message M à Jacques.

- Paul utilise la clé publique $C_{jacques}$ de Jacques pour crypter le message qu'il transmet à Jacques.
- Jacques déchiffre le message reçu en lui appliquant $D_{jacques}(C_{jacques}(M))$, personne d'autre n'est capable de déchiffrer le message $C_{jacques}(M)$.

L'algorithme du MIT

1. choisir 2 nombres premiers, p et q, chacun plus grands que 10^{100}

2. calculer n=p.q et z=(p-1)(q-1)

3. choisir un nombre d premier avec z

4. chercher un nombre e tel que e.d=1(modz)

Découper le texte en une suite de blocs de telle sorte que chaque bloc de texte en clair M soit un nombre tel que 0<=M<n

pour crypter : $C = M^e \pmod n$ la clé publique = (e,n)

pour déchiffrer : $D = C^d \pmod n$ la clé privée = d

La sécurité de la méthode réside dans la difficulté à décomposer de très grands nombres en facteurs premiers.

Ex.:

p=3, q=11, n=33, z=20, d=7, e=3

Texte en clair	N	I	C	E
M	14	9	3	5
M^3	2744	729	27	125
Texte chiffré $C=M^3 \pmod{33}$	5	3	27	26
C^7	78125	2187	-	-
$C^7 \pmod{33}$	14	9	3	5
Texte en clair	N	I	C	E

IV Intégrité

Contrôle de la validité des données

Contrainte d'intégrité

Une contrainte d'intégrité est une assertion qui doit être vérifiée par des données à des instants déterminés.

- Les contraintes d'intégrité permettent de préciser davantage la partie intentionnelle (sémantique) de la base de données.

- Une base de données est cohérente vis à vis des contraintes qui sont exprimées, si ces contraintes sont respectées par les données de la base.

Gestion des contraintes d'intégrité

Expression des contraintes

L'écriture des différents types de CI est prévue dans de nombreux langages

Par exemple la clause CHECK de SQL/ORACLE

Vérification des contraintes

Les CI sont vérifiées lors des mises à jour (en fin de transaction)

C'est très coûteux en temps machine, il est essentiel de pouvoir vérifier ces contraintes de manière efficace

Violation des contraintes

Une mise à jour qui provoque la violation d'une CI est refusée

L'intégrité de la base de données est préservée par le SGBD

Utilisation des déclencheurs (trigger)

Un **déclencheur** (*trigger*) permet de définir un ensemble d'actions qui sont déclenchées automatiquement par le SGBD lorsque des mises à jour sont effectuées.

- Les actions sont enregistrées dans la base et plus dans les programmes d'application

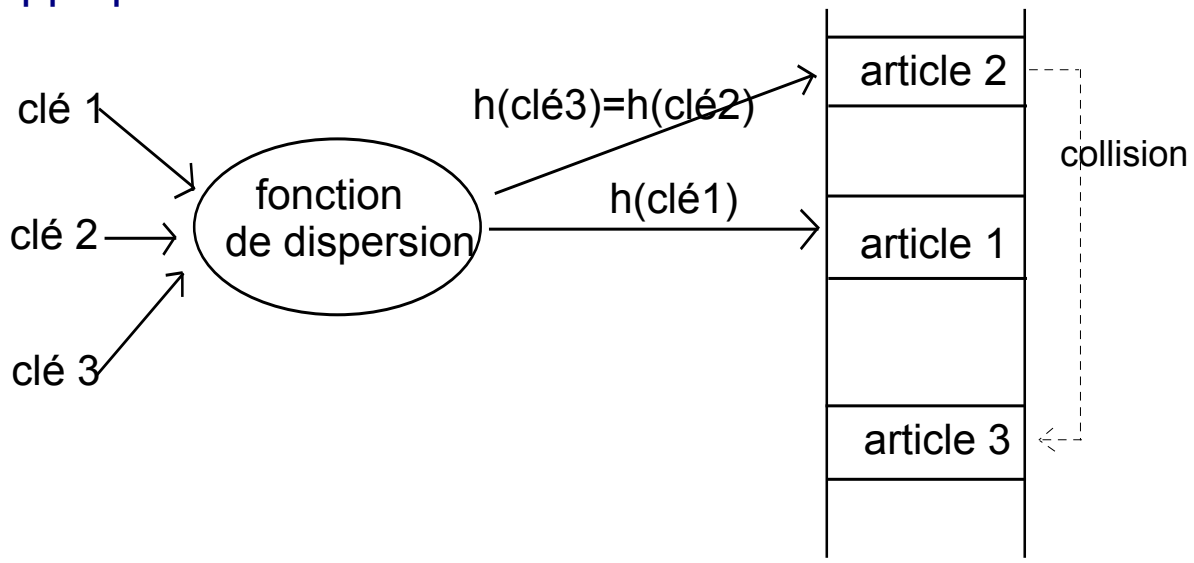
- Cette notion n'est pas encore spécifiée dans SQL 2

- Elle est présente dans les principaux SGBD (Oracle, Sybase, DB2, SQL Server)

V Méthodes d'accès

Méthode d'accès par dispersion

L'adresse relative d'un article (ou d'un paquet contenant l'article) est obtenu par une fonction de hachage appliquée à la clé



Traitement des collisions lorsqu'un paquet est plein

- *adressage ouvert*
  prendre le premier paquet suivant ayant de la place libre
- *chaînage*
  chaîner un paquet de débordement au paquet plein
- *rehachage*
  appliquer une deuxième fonction de hachage

Avantages

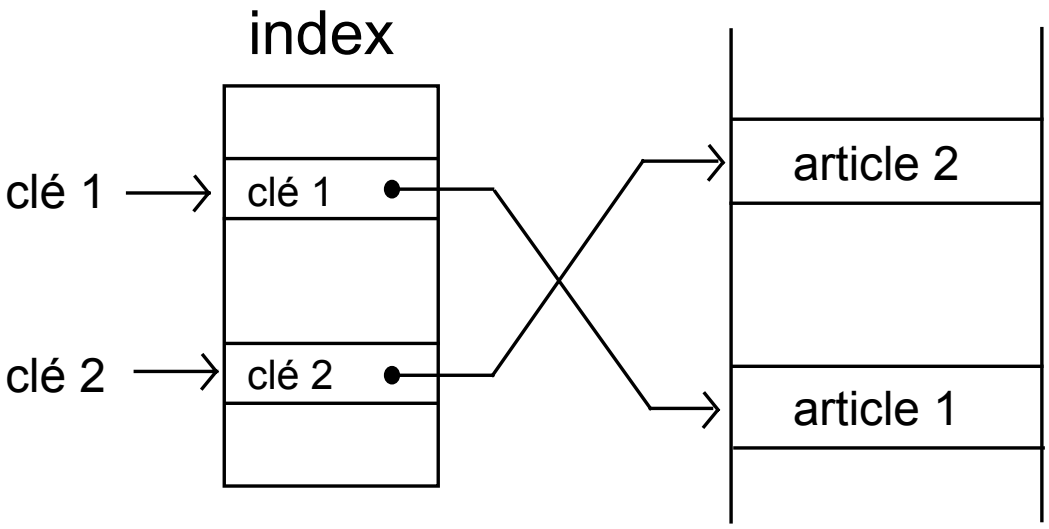
rapidité si l'on parvient à réaliser un faible taux de collisions

Inconvénients

pas d'accès séquentiel trié dans l'ordre des clés

Méthodes d'accès par indexage

L'adresse relative d'un article (ou d'un paquet contenant l'article) est recherchée à partir de la clé dans une table d'index



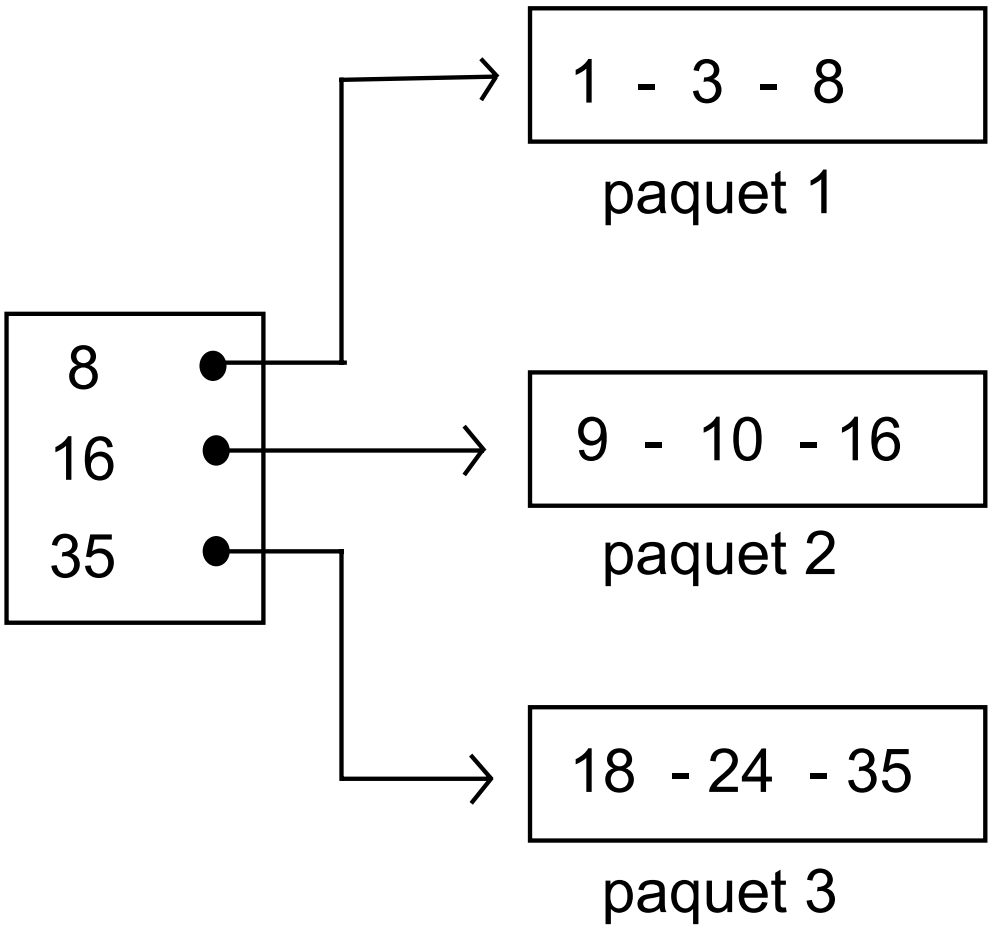
Principales méthodes d'accès indexées

ISAM

ARBRE B

elles se distinguent par le mode de placement des articles et par l'organisation de l'index

ISAM (Indexed Sequential Acces Method d'IBM)

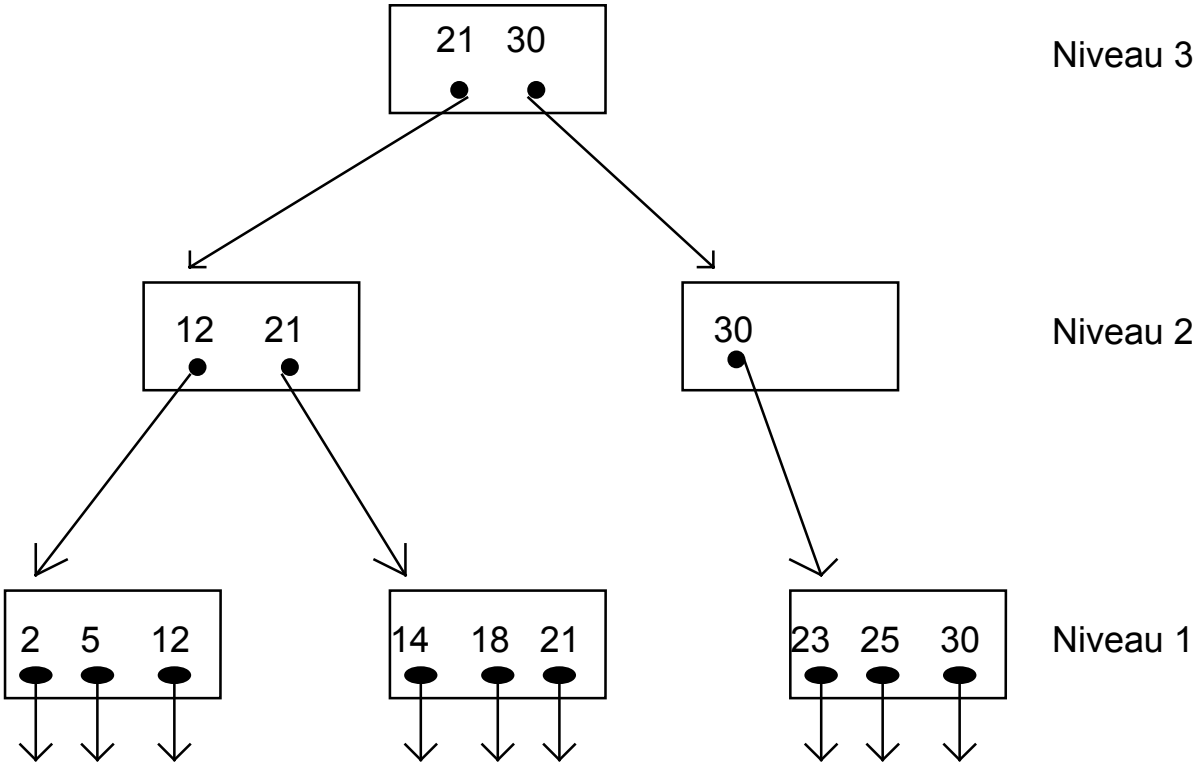


Les articles sont rangés dans des paquets de taille fixe par ordre croissant des clés

Chaque paquet correspond à une entrée en index contenant le doublet :
(plus grande clé du paquet, adresse relative du paquet)

ARBRE B

L'index est hiérarchisé en plusieurs niveaux



L'index est composé de paquets de clés

2 types de pointeurs :

- pointeur interne permettant de représenter l'arbre
- pointeur externe sur l'adresse relative d'un article

Les paquets de niveau k+1 contiennent les plus grandes clés des paquets de niveau k