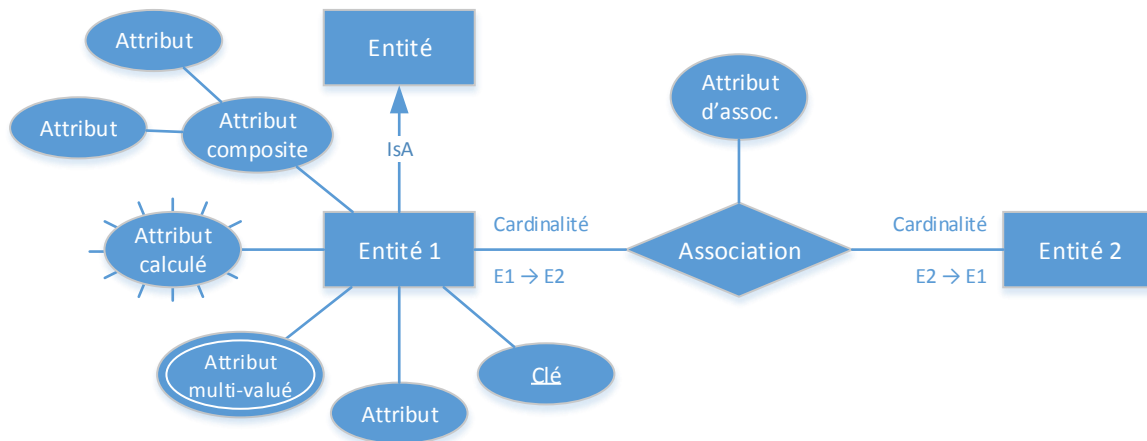


I. Modèle Entité-Association



II. Décomposition

Dépendance fonctionnelle : (X détermine $Y \Leftrightarrow X \rightarrow Y$) : même $X \Rightarrow$ même Y (Ex : $\hat{m}$ ID $\Rightarrow$ $\hat{m}$ nom)

Dépendance fonctionnelle élémentaire : $X \rightarrow A$ tel que $A \not\subseteq X$ et $\nexists X' \subseteq X \mid X' \rightarrow A$

Propriétés de la DF :

- **Réflexivité** : $Y \subseteq X \Rightarrow X \rightarrow Y$
- **Augmentation** : $X \rightarrow Y \Rightarrow XZ \rightarrow YZ$
- **Transitivité** : $X \rightarrow Y \wedge Y \rightarrow Z \Rightarrow X \rightarrow Z$
- **Union** : $X \rightarrow Y \wedge X \rightarrow Z \Rightarrow X \rightarrow YZ$
- **Pseudo-transitivité** : $X \rightarrow Y \wedge YW \rightarrow Z \Rightarrow XW \rightarrow Z$
- **Décomposition** : $X \rightarrow Y \wedge Z \subseteq Y \Rightarrow X \rightarrow Z$

Fermeture transitive F^+ : Union de F (ensemble de DF) et des DF déduites par transitivité.

Couverture minimale F_{CM} : Ensemble de DF élémentaires permettant de générer toutes les autres.

Clé : Groupe minimal d'attribut d'une relation permettant d'identifier un tuple de manière unique.

Décomposition de R en $\{R_i\}$: préserve les DF si $F_R^+ = \cup F_{R_i}$

III. Formes normales

- **1FN** : Tout attribut a une valeur atomique.
- **2FN** : Tout attribut n'appartenant pas à une clé ne dépend pas que d'une partie de cette clé.
- **3FN** : Tout attribut n'appartenant pas à une clé ne dépend pas d'un attribut non clé.
- **BCNF** : Les seules DF sont non triviales et sont celles dans lesquelles une clé détermine 1 ou plusieurs attributs.

IV. Concurrency

- **BD cohérente** si contraintes d'intégrités vérifiées
- **Transaction** : ensemble d'opération permettant passage entre 2 états cohérents.
Respecte les propriétés ACID (Atomicité, Cohérence, Isolation, Durabilité).

1. Problèmes possibles

- Perte d'opération (override)
- Perte de cohérence
- Lecture impropre (ex : transaction annulée)
- Sortie écran impropre
- Lecture non reproductible dans trans. (~~isolation~~)

2. Opérations

X : donnée / A : variable

- LIRE(X, A)
- ECRIRE(X, A)
- AFFICHER(A)
- LOCKW(X) verrou exclusif
- LOCKR(X) verrou partagé
- UNLOCK(X)

3. Verrouillage en 2 phases

- Verrou avant toute lecture/écriture
- Verrouillages puis déverrouillages (pas de mélange)
- Solution au deadlock : 1^{ère} transaction lancée prioritaire annule la 2^{ème}

V. Organisation physique des données

1. Placements

- Séquentiel non-trié
- Séquentiel trié
- Indexé
- Par hachage

2. Caractéristiques

- Mono/multi-attribut
- Clé primaire/non primaire
- Statique/Dynamique
- Comparaison de clé/Calcul

3. Accès

Méthodes de recherche

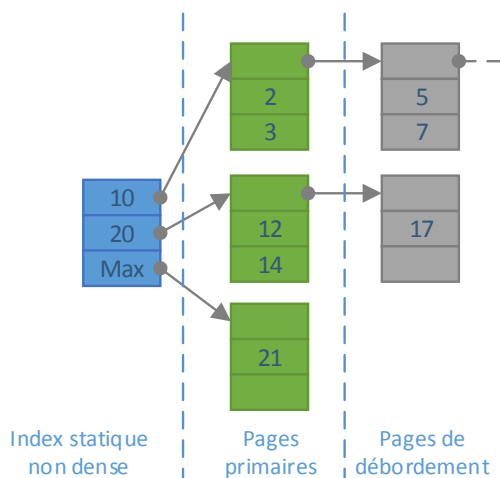
- Séquentiel
- Sélectif (direct)

4. Placements séquentiels

- **Séquentiel non-trié** : placé par ordre d'insertion sans tri
- **Séquentiel trié** : placé selon un index de tri ordonné / décalage des données

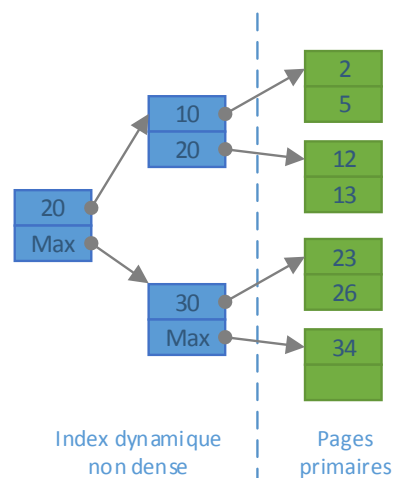
5. Placement indexé

a. Statique



Performant si pas de débordement
Insertion coûteuse si débordement
Nécessite réorganisation courante

b. Dynamique



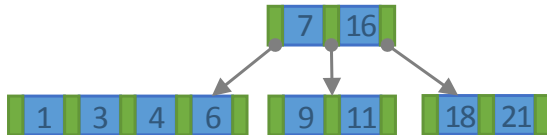
Insertion coûteuse si réorganisation nécessaire
mais réorganisation « automatique »

c. Dynamique : Arbre-B

- Un nœud contient k clé triées $m \leq k \leq 2m$ ($1 \leq k \leq 2m$ pour la racine)
- L'arbre est équilibré (différence de hauteur de 1 max)
- Un nœud non terminal à $k + 1$ fils t.q. les clés d'un fils ont des valeurs comprises entre les clés l'encadrant dans le père (strict).

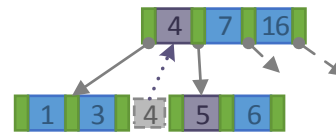
Nb clé max : $2(m + 1)^h - 1$ / Nb clé min $(2m + 1)^{h+1} - 1$

Exemple d'arbre d'ordre 2



Insertion (5) : nœud plein

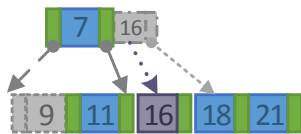
Eclatement autour de la clé médiane qui remonte, récursivement. Peut créer une nouvelle racine.



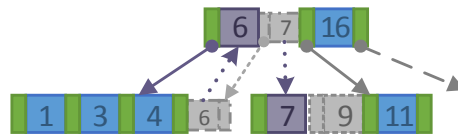
Suppression feuille : nombre de clé insuffisant

Combinaison avec voisin, descente de la clé médiane

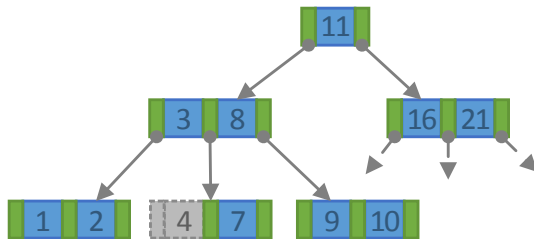
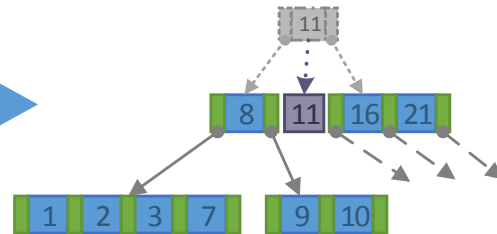
Normal (9, combiné à droite)



Débordement (9, g.) : cf insertion (découpage + remonté méd.)

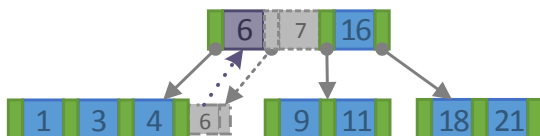


Parent trop petit : diminution de la hauteur (comb. du parent avec son voisin + descente med.)



Suppression non-feuille (7)

Remontée du plus grand des plus petits.



Arbre B+

Comme l'arbre B mais les clés des nœuds non-terminaux sont dupliquées dans leur fils gauche respectif. Toutes les clés sont présentes dans les feuilles.



d. Hachage

On utilise une fonction de hachage sur la donnée pour déterminer sa position dans un index et donc déterminer quel fichier la contient. Il peut être extensible (on change la taille de l'index pour avoir plus de hashes différents et avoir moins de collision par hash).

6. Chemins d'accès

L'accès est toujours facilité sur l'attribut de placement (accès primaire). Possibilité de créer un autre index pour avoir un chemin d'accès secondaire sur autre attribut (accès secondaire).

a. Les types de clés

- **Clé ... :** Attribut ou groupe d'attribut qui...
- **... primaire relation^{elle} :** identifie un tuple de manière unique
- **... de placement :** détermine placement, Clé d'accès primaire.
- **...d'accès secondaire :** Chemin d'accès secondaire. Index liant clé secondaire et clé primaire

b. Temps d'accès

- **Balayage :** $T_{\text{accès moy.}} = N_{\text{blocs}} \times T_{\text{accès bloc}} / N_{\text{art.recherchés}}$
- **Index :** $T_{\text{accès moy.}} = N_{\text{niveaux index}} \times T_{\text{accès bloc}}$

VI. Droits d'accès

Interrogation	Mise à jour	Création	Administration
SELECT	INSERT, UPDATE, DELETE	CREATE TABLE	contraintes d'intégrité, index, accorder droits

- **Accorder droits :** GRANT <droits> ON <relation> TO <usagers> [WITH GRANT OPTION]
- **Supprimer droits :** REVOKE <droits> ON <relation> TO <usagers> [WITH GRANT OPTION]
transfert du droit possible

VII. Contraintes d'intégrité

Assure la cohérence logique de la base.

```
CREATE TABLE table1 (  
  Attr1 TYPE [CONTR1 [CONTR2, [...]]],  
  Attr2 TYPE [CONTR1 [CONTR2, [...]]]  
  ...  
)
```

1. Structurelle

Structure du modèle cohérente

3. Intra-relation

Dans la relation

- **Clé :** PRIMARY KEY
- **Domaine :** CHECK(attr > x)
CHECK(attr IN (... , ... , ...))
- **Non nullité :** NOT NULL
- **Unicité clé :** UNIQUE
- **DF :** A → B
- **Temporelle :** modif dans le temps
- **Agrégat :** sur le res. d'une fct d'agrégat

5. Trigger

Fonction déclenchée dans certaines conditions suite à un évènement (insertion par ex.) pour vérifier l'intégrité de façon plus complexe.

2. Comportementales

Lié aux applications

4. Inter-relation

Entre relations

- **Référentielle :** REFERENCES table
(clé étrangère) [ON DELETE CASCADE]
- **Inclusion :** $T_1.A \subset T_2.B$
- **Générale**