Structures de fichiers (SFSD)

Introduction
aux Bases de Données

- opérations sur les fichiers -

École nationale Supérieure d'Informatique (ESI)

Notions de Bases de données

- SGBD (Système de Gestion de Bases de Données)
  - « grand » volume de données liées entre elles (= **structure**)
  - ensemble de programmes qui gèrent ces données (= **traitement**)
- Objectifs
  - efficacité (très important)
  - sécurité
  - facilité d'utilisation

Intérêts des SGBD

- Redondance et inconsistance des données
  - certaines info se trouvent sur plusieurs fichiers
- Difficulté d'accès aux informations non prévues
  - nécessité d'écrire de nouveaux prog. d'accès
- Dépendance : rep. interne / Applications
  - changement de structure -> re-programmation des App
- Atomicité et pb de concurrence
  - erreur, pannes, accès concurrents -> introduisent des inconsistances

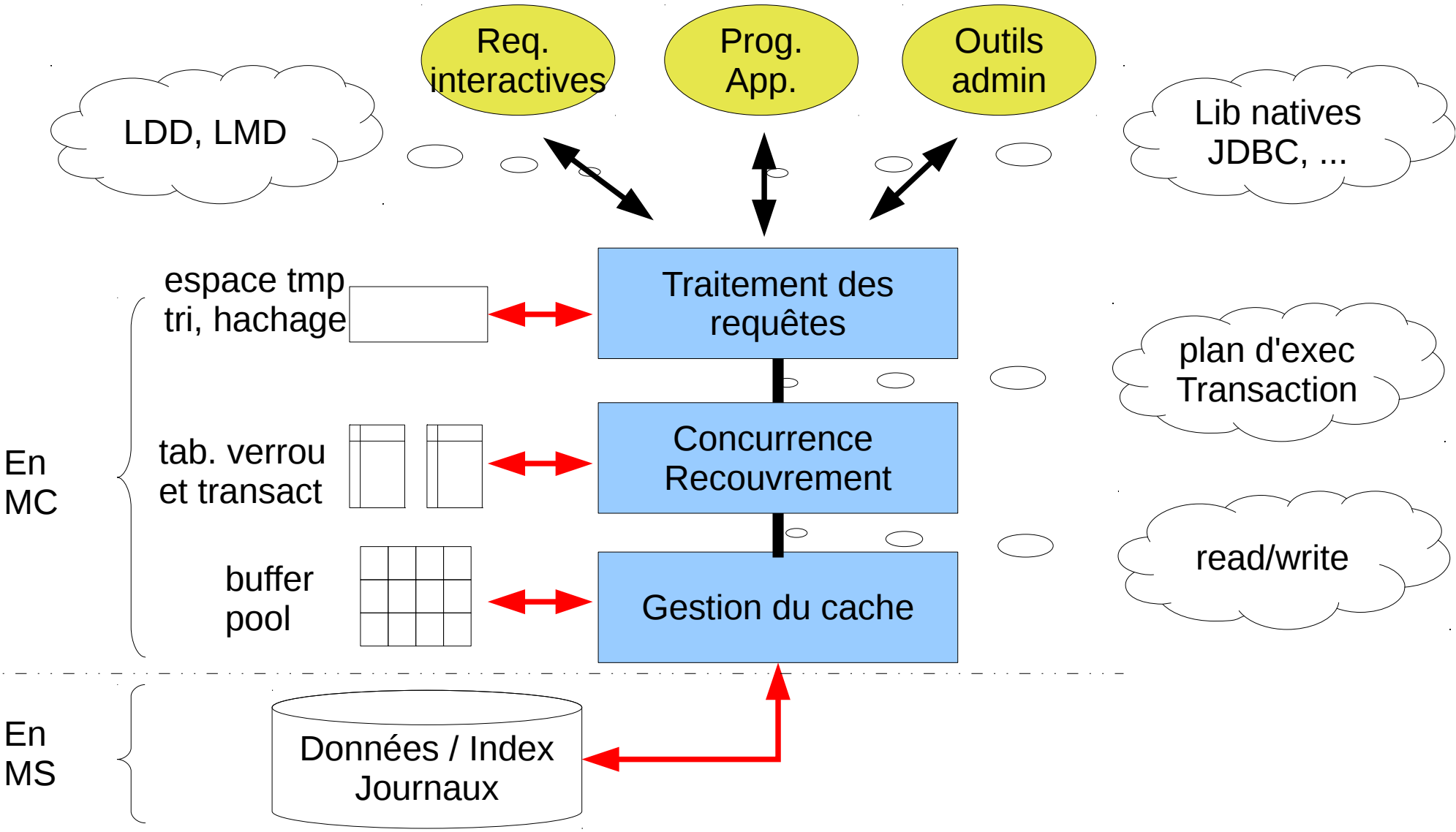
Modèles de données

- Définition

  Outils conceptuels pour décrire les données, les liens, la sémantique et les contraintes.

- Exemples de modèles
  - Le modèle Entités-Associations
  - Le modèle Relationnel
  - Le modèle Objet
    - Le modèle Relationnel-Objet
  - Anciens modèles:
    - Hiérarchique, Réseau (CODASYL)

Architecture du SGBD



Le modèle relationnel

- BD = ensemble de tables (des fichiers)
- Table T(A1:D1, A2:D2, … An:Dn)
  - sous-ens du produit cartésien des domaines de ses constituants. $T \subseteq \{D1xD2x...Dn\}$
  - Une table est donc une Relation

Par	Enf
a	c
b	c
e	d

Une relation binaire

Pere	Mere	Enf
a	b	c
e	f	d
e	f	g

Une relation ternaire

NProd	NFour	Prix	Date
P54XV	F761	13.65	16/03/07
P03AT	F761	23.50	27/01/07
P54XV	F103	13.55	03/12/06

Une relation n-aire

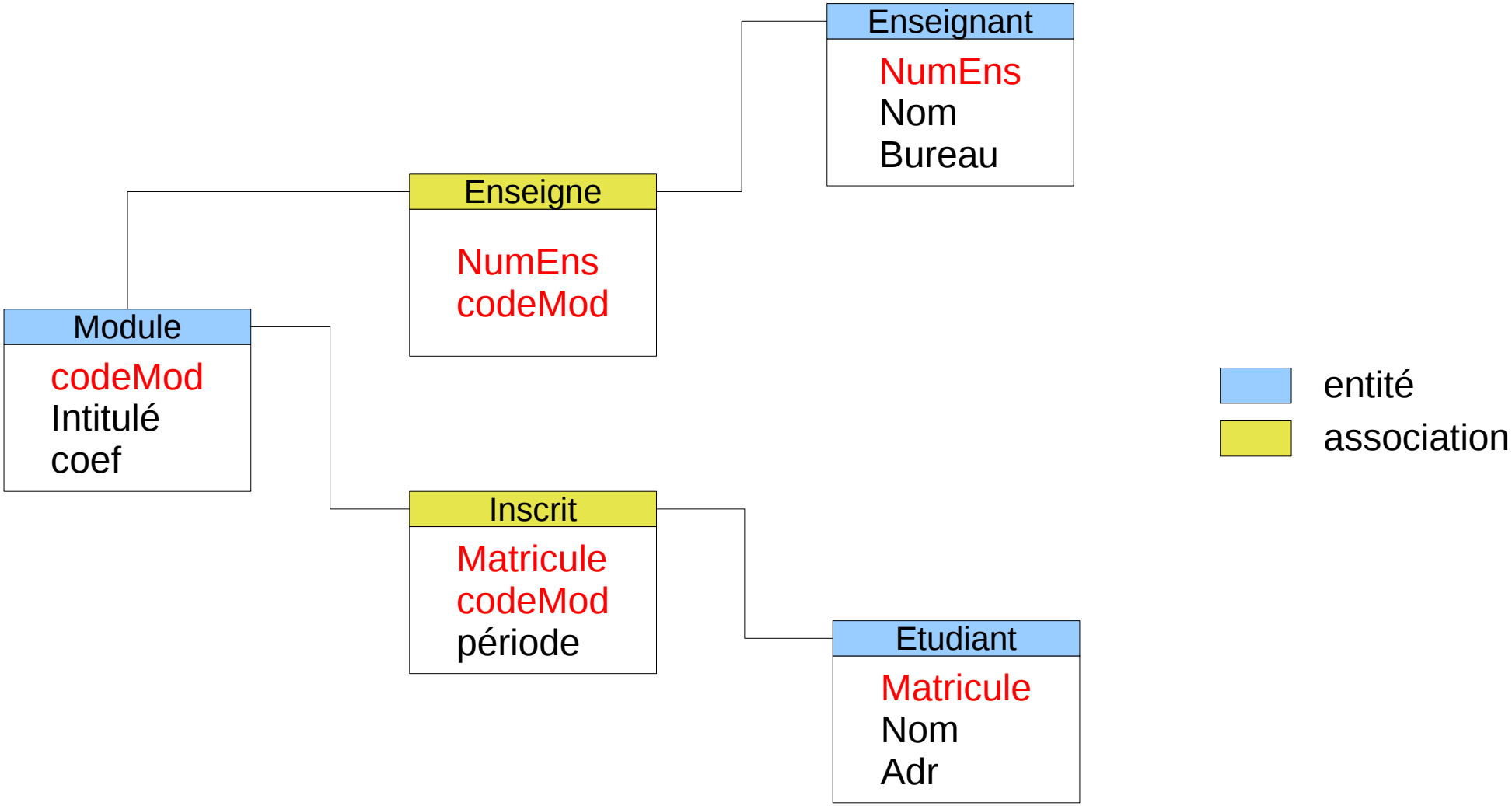
tuples

Conception de schémas

- Il y a des tables qui représentent des entités avec leurs propriétés
  - Patient(NumP, Nom, Adr)
  - Pathologie(code, description)
- Il y a des tables qui représentent des associations entre entités
  - Malade(NumP, code, date)

    Cette relation **associe** un patient (NumP) avec une pathologie (code)

Exemple


Enseignant
NumEns
Nom
Bureau
Enseigne
NumEns
codeMod
Module
codeMod
Intitulé
coef
Inscrit
Matricule
codeMod
période
Etudiant
Matricule
Nom
Adr
entité
association

Algèbre relationnelle

- Langage de requêtes procédural
  - On doit spécifier comment trouver le résultat
- Chaque opération a 1 ou 2 relations en entrée et 1 relation en sortie
- Une requête est une expression (composition) d'opérations algébriques

Quelques opérateurs relationnels

opérateurs unaires

- Sélection ou Restriction

  σ (cond / R)

- Projection

  Π (liste_attr / R)

- Renommage

  r (R1 / R2)

opérateurs binaires

- Union

  R1 $\cup$ R2

- Différence

  R1 - R2

- Produit cartésien

  R1 x R2

Exemple de requêtes

Soient 2 tables relationnelles
- Produit(codeP, NomP, description)
- Compose(p1, p2, qte)
  // le produit 'p1' est composé de 'qte' sous-prod 'p2'

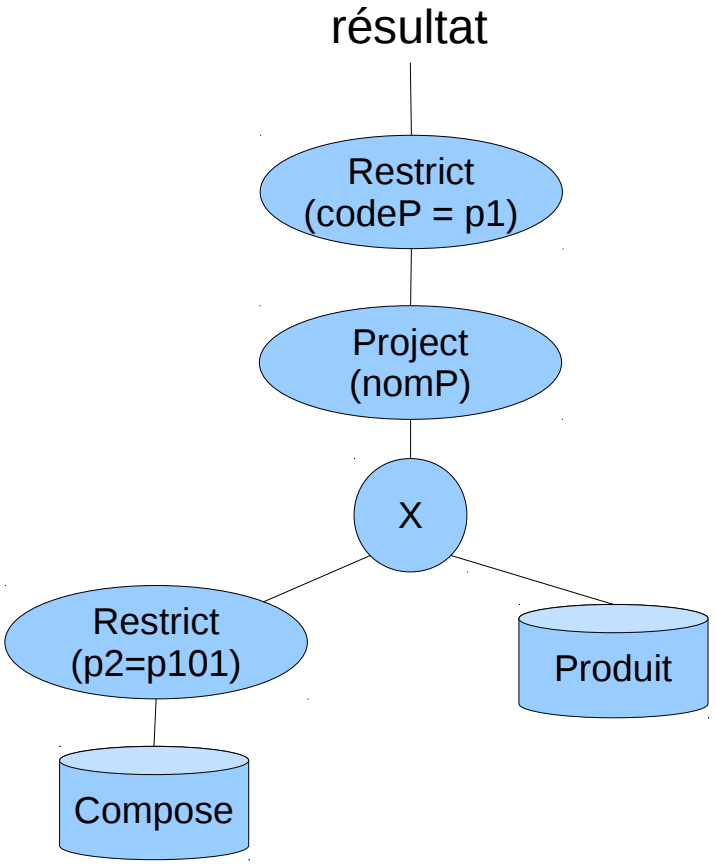
Donner le nom des produits composés par le sous-produit de code 'P101'

R1 = σ (p2='P101' / Compose)

R2 = Produit X R1

R3 = σ (codeP=p1 / R2)

Résultat = ∏ (NomP / R3)



Autres opérateurs algébriques

- Intersection (R et S de même schéma)

  $R \cap S = R - (R - S)$

- Jointures (naturelle – cnd implicite, théta – cnd explicite)

  $R \otimes_{(cnd)} S = \sigma (cnd / R \times S)$

- Division (schéma de S $\subset$ dans schéma de R)

  $R_{(a1,a2,..b1,b2,...)} \div S_{(b1,b2,...)} = T_{(a1,a2,..)}$

  $= \Pi(a1,a2,.../ R) - \Pi(a1,a2,.../ ((\Pi(a1,a2,.../ R) \times S) - R))$

  un tuple t est dans T ssi
  - t est dans Π(a1,a2,.../ R) et,
  - pour chaque tuple ts dans S, il existe un tuple tr dans R :
    tr[b1,b2,...] = ts et tr[a1,a2,...] = t

Le langage SQL

SELECT exp1, exp2, ...
   FROM R1, R2, ...
   WHERE <Cond>

= Π(exp1, exp2, ... / σ (Cond / R1 x R2 x ...))

Exemples:

SELECT NomP FROM Produit WHERE codeP='P653';

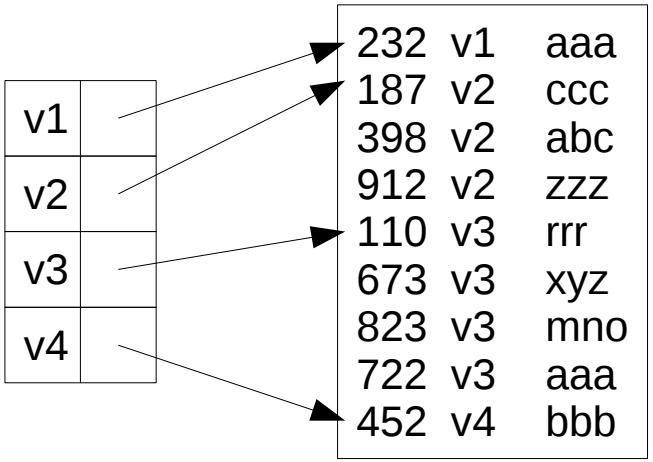
SELECT NomP, qte
   FROM Produit, Compose
   WHERE codeP=P2 AND P1='P326';

Stockage

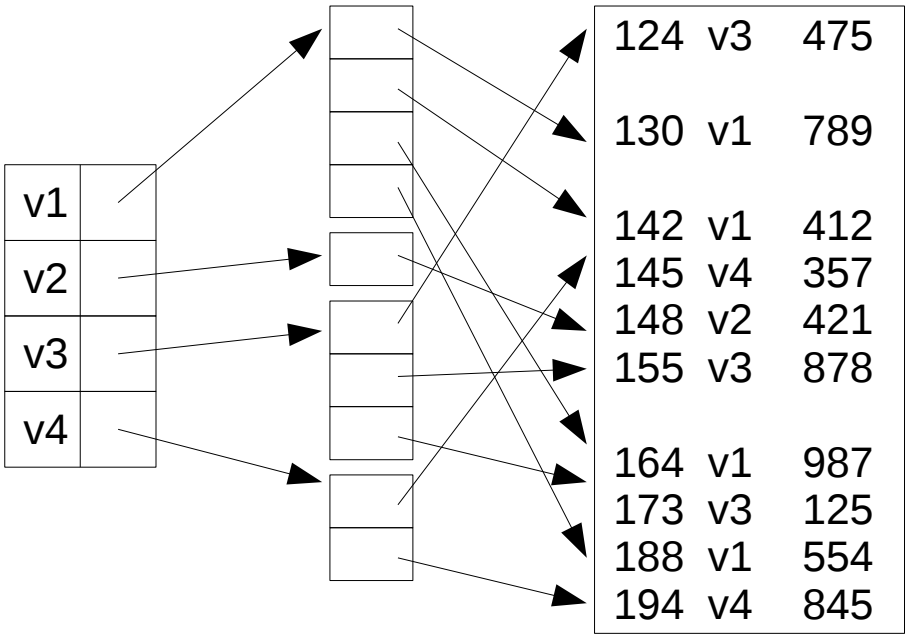
- Fichier = ensemble de blocs d'enregistrements
- un enregistrement peut être de taille fixe ou variable (longueur de champs, nombre de champs)
- organisation de fichier
  - non ordonné (heap file)
  - ordonné (sur un ou + attributs) : fichier séquentiel
- Clustering
  - rassembler dans le même bloc, les enreg susceptibles d'être traités ensembles.

Index

- structuré en tables ou arbres (B-Arbres)
- Primaire (dense ou non dense)
  si le fichier est ordonné suivant l'attribut indexé
- Secondaire (dense)
  si le fichier n'est pas ordonné



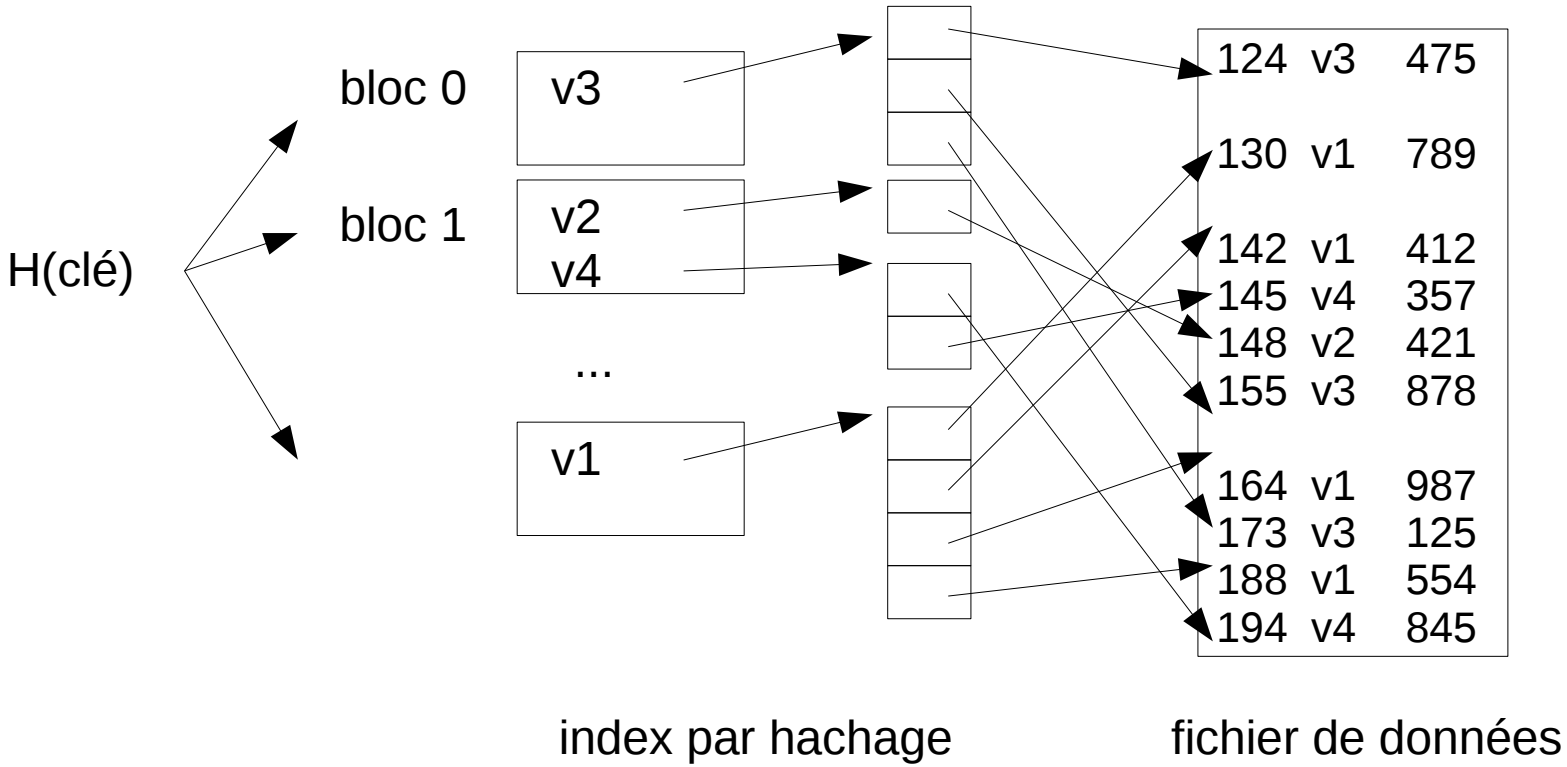
index primaire dense (contient toutes les valeurs)



index secondaire

Hachage

- On peut construire une table de hachage jouant le rôle d'index sur un fichier de données existant

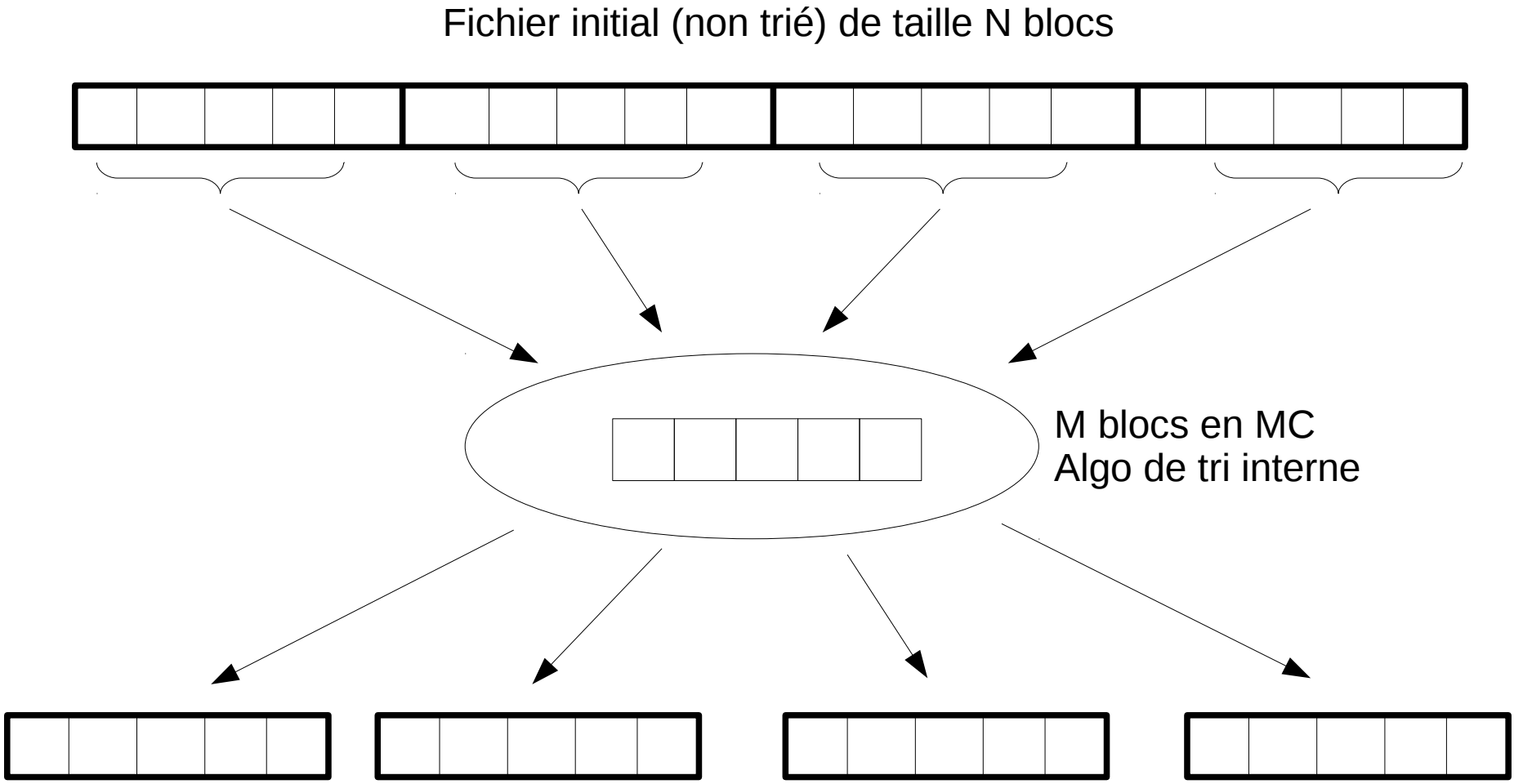


Implémentation des Opérateurs

- Sélection (avec et sans index)
- Tri « par fusion » (en 2 phases)
- Jointure
  - boucles imbriquées
  - « par fusion »
  - « par hachage »
- Autres opérateurs (intersection, différence, …)
  - Utilisent des algorithmes similaires à ceux de la jointure

Tri par fusion (externe)

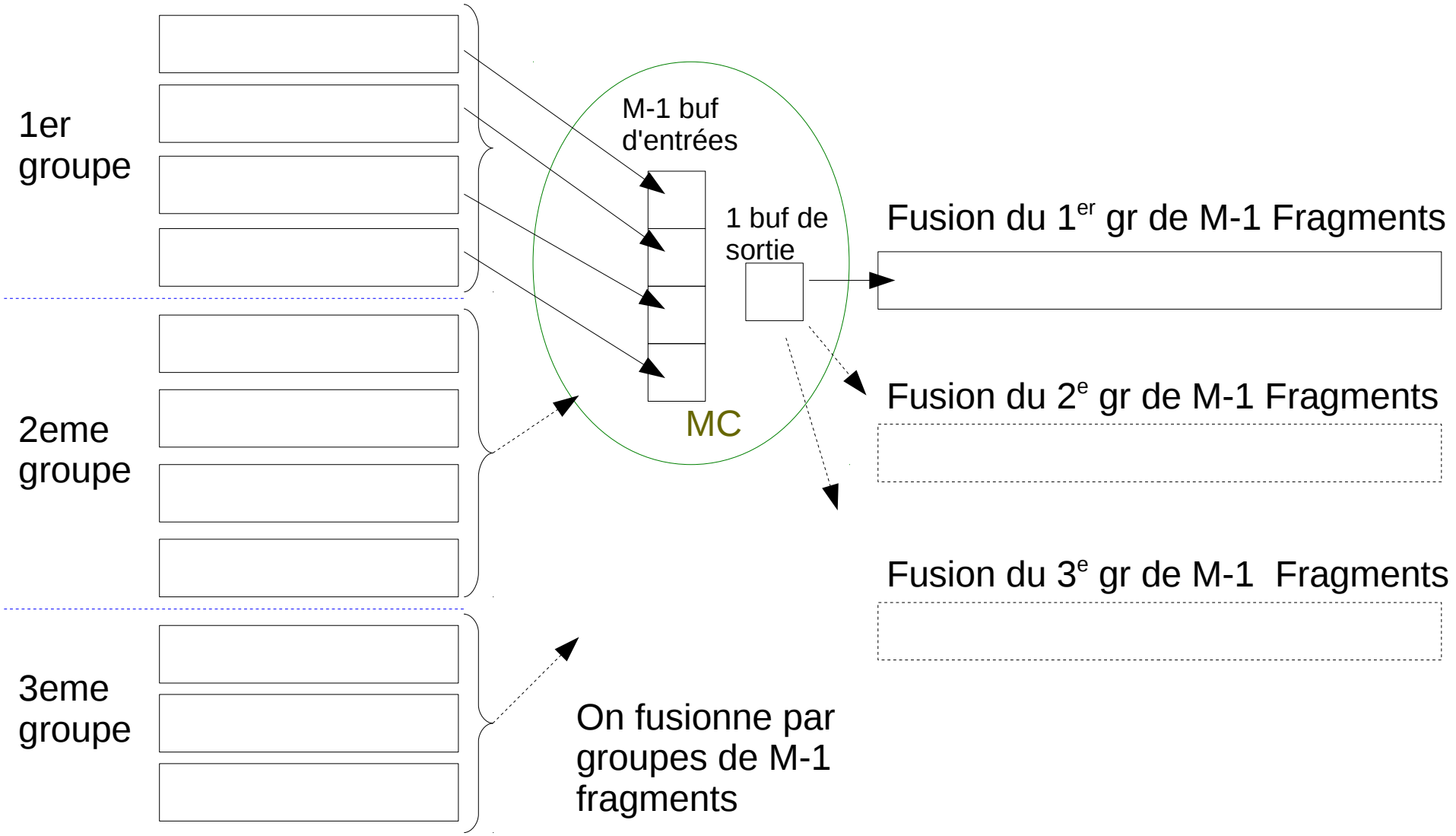
- première phase -



[N/M] Fragments initiaux triés, de taille M blocs chacun

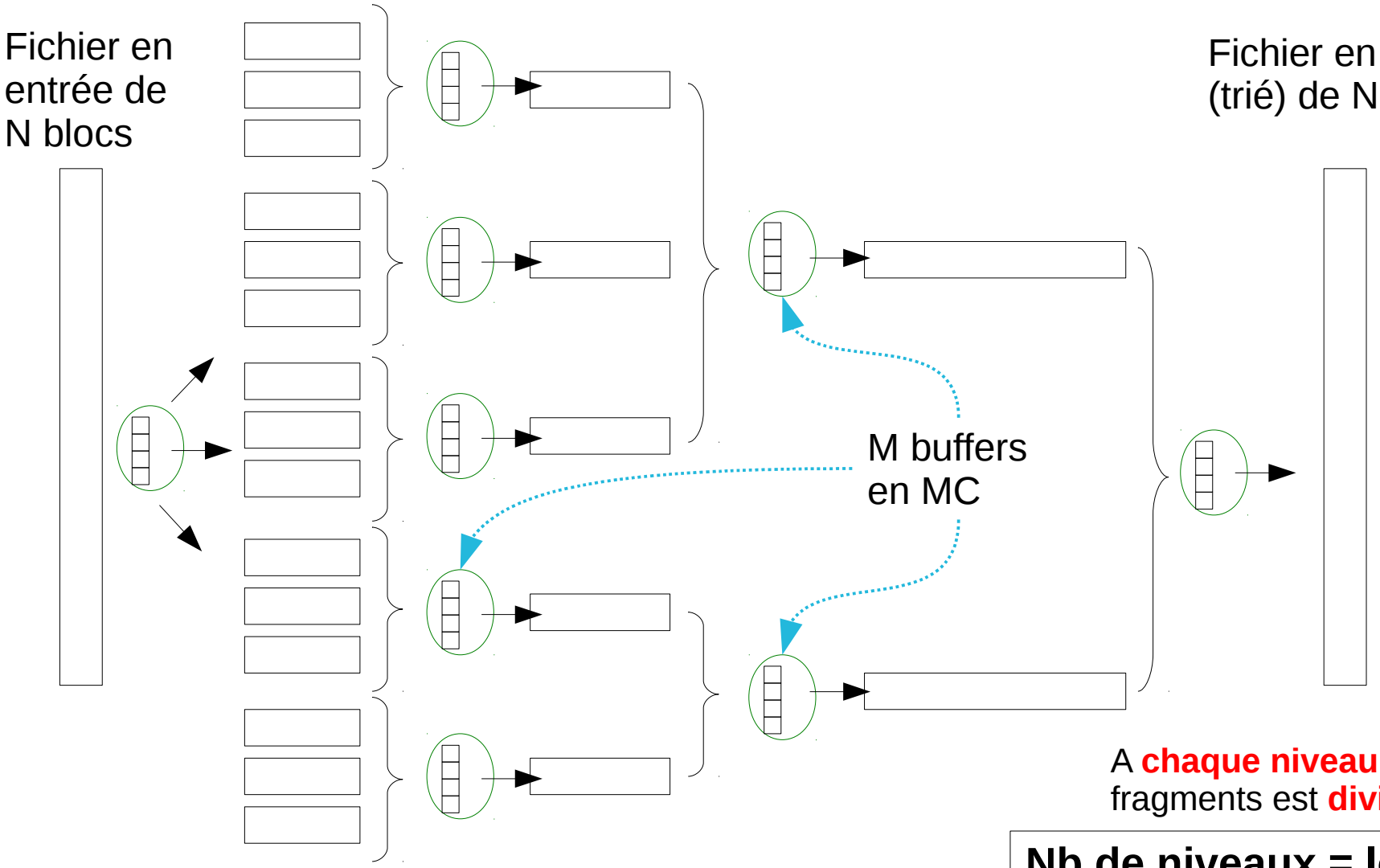
Tri par fusion (externe)

- Deuxième phase -



Tri par fusion (externe)

- Schéma général -



A **chaque niveau** le nb de fragments est **divisé par M-1**

Nb de niveaux = $\log_{M-1}[N/M]$

à chaque niv, il faut 2N accès

Fragmentation **Fusion**

Jointure par boucles imbriquées

R (Relation externe : N_R blocs) x S (Relation interne : N_S blocs)

- Utiliser le **max d'espace** mémoire pour la **relation externe** (la plus petite)
  M-2 buffers pour R, 1 buffer pour S et 1 buffer pour écrire le résultat.
- Lire **R** par fragments de **M-2 blocs**. Pour chaque fragment de **R**, lire **S**

Pour chaque **fragment Fr** de **R** (lecture de M-2 blocs de **R**)

Pour chaque **bloc Bs** de **S** (lecture d'un bloc de **S**)

Pour chaque tuple tr dans Fr

Pour chaque tuple ts dans Bs

Si la condition de jointure est vérifiée entre tr et ts
rajouter la concaténation tr.ts au résultats B
Si B est plein, le vider sur disque

- Coût de l'opération (sans compter l'écriture du résultat) :

  S est lue $[N_R/(M-2)]$ fois. **R** est lue une seule fois

  = $\mathbf{N_R + N_S\ [N_R/(M-2)]}$ accès disque