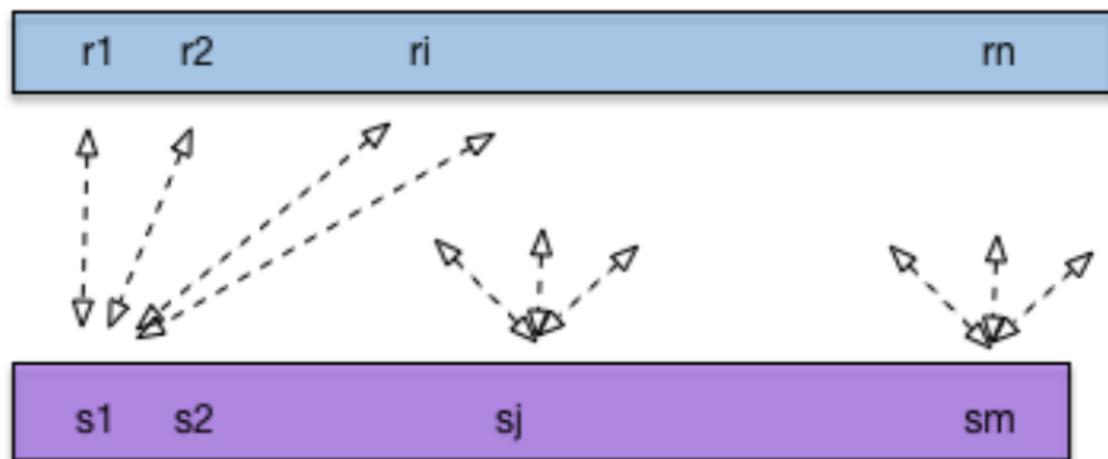


Jointures par boucles imbriquées

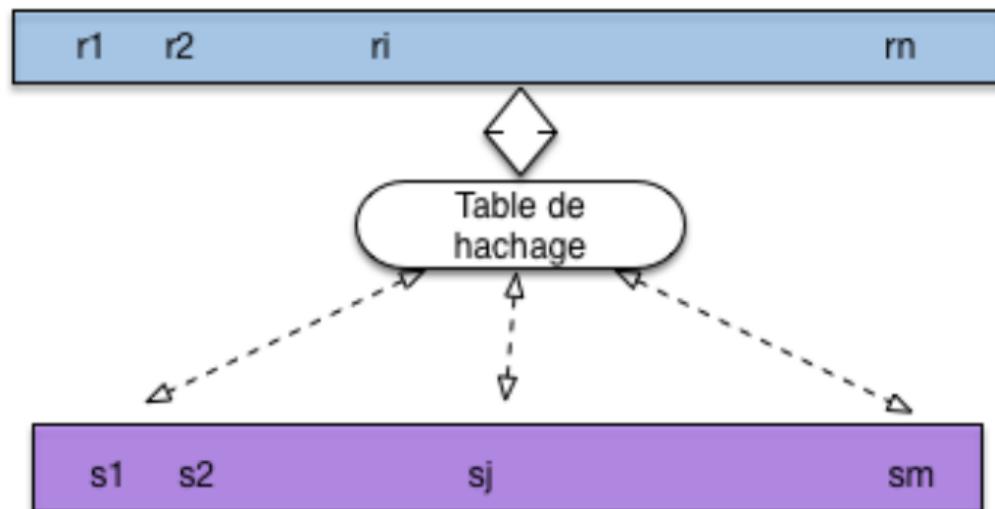
Pas d'index ? La méthode de base est d'énumérer **toutes** les solutions possibles.



Coût quadratique. Acceptable pour deux petites tables.

Jointures par hachage

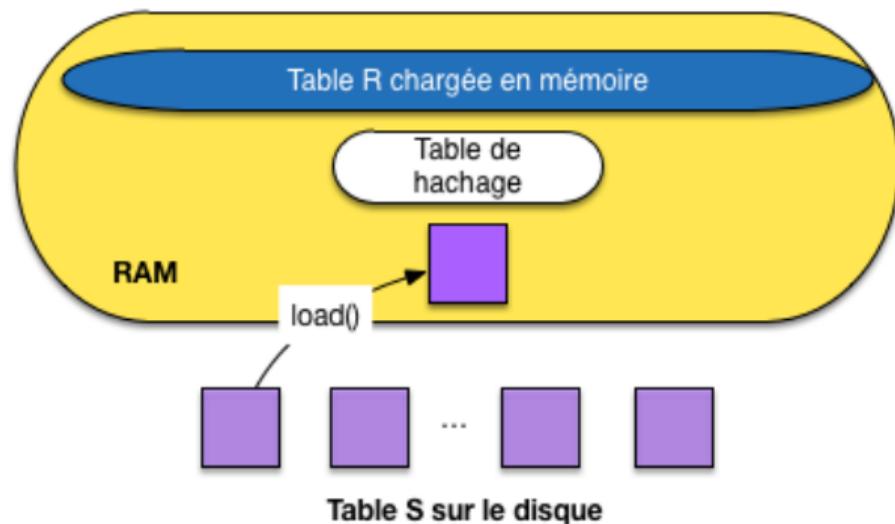
Meilleure solution : construire une table de hachage sur une des tables.



Evite les $O(n^2)$ comparaisons. Appelons cette méthode **JoinList**.

Mémoire insuffisante ?

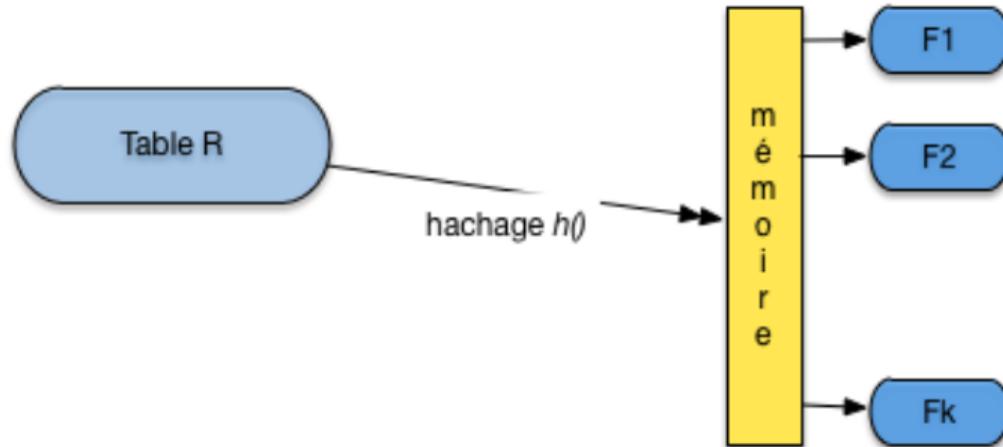
Essayons de placer **une** des deux tables en mémoire.



On charge l'autre bloc par bloc ; on applique **JoinList**.
Une seule lecture de chaque table suffit.

Et quand *aucune* table ne tient en mémoire ?

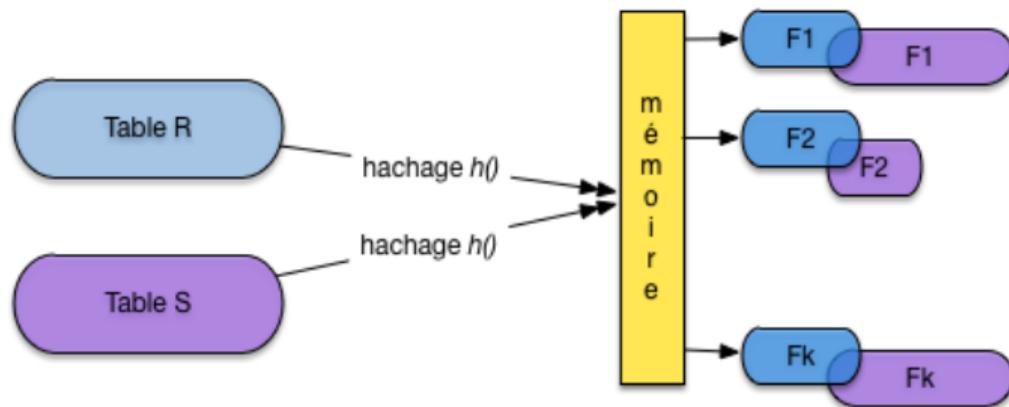
On hache la plus petite des deux tables en k fragments.



Essentiel : les fragments doivent tenir, chacun, en mémoire.

Et quand *aucune* table ne tient en mémoire ?

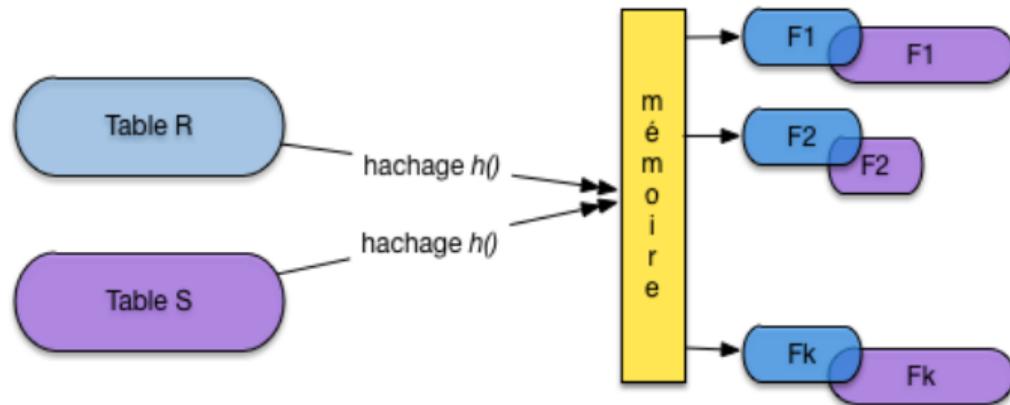
On hache la seconde table, avec la même fonction $h()$, en k autres fragments.



Cette fois, on n'impose pas la contrainte que les fragments tiennent en mémoire.

Et quand *aucune* table ne tient en mémoire ?

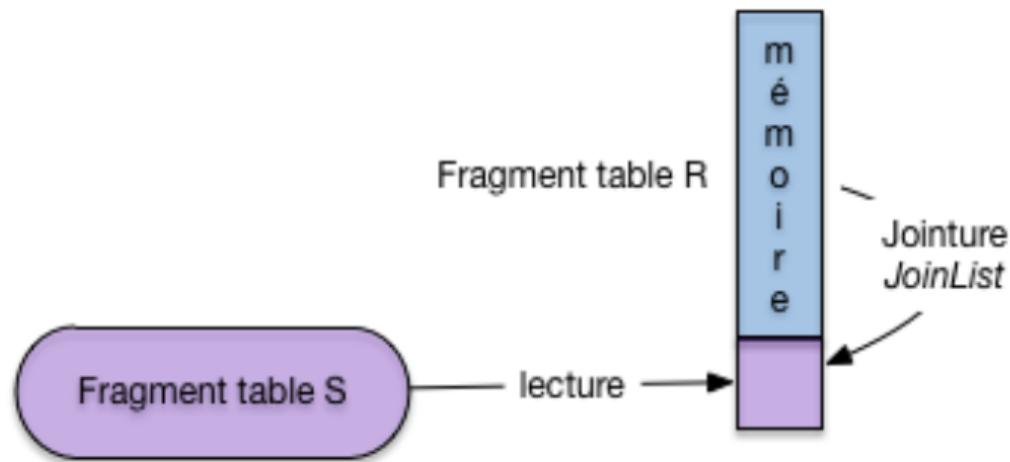
On effectue la jointure sur les paires de fragments $(F_1^R, F_1^S), (F_2^R, F_2^S), (F_k^R, F_k^S),$



Propriété : Deux nuplets r et s doivent être joints si et seulement si ils sont dans des fragments associés.

Illustration : phase de jointure

On charge F_R^i de R en mémoire ; on parcourt F_S^i de S et on joint.



Déjà vu ? Oui : jointure par boucles imbriquées quand une table tient en mémoire.

Résumé : la jointure

Un opérateur potentiellement coûteux. Quelques principes généraux :

- Si **une** table tient en mémoire : jointure par boucle imbriquées, ou hachage.
- Si **au moins un** index est utilisable : jointure par boucle imbriquées indexée.
- Si une des deux tables beaucoup plus petite que l'autre : jointure par hachage.
- Sinon : jointure par tri-fusion (non présenté).

Décision très complexe, prise par la système en fonction des statistiques.

Résumé : la jointure

Un opérateur potentiellement coûteux. Quelques principes généraux :

- Si **une** table tient en mémoire : jointure par boucle imbriquées, ou hachage.
- Si **au moins un** index est utilisable : jointure par boucle imbriquées indexée.
- Si une des deux tables beaucoup plus petite que l'autre : jointure par hachage.
- Sinon : jointure par tri-fusion (non présenté).

Décision très complexe, prise par la système en fonction des statistiques.

Merci !