

## Tables

```
CREATE TABLE Produit(  
  NPro INT,  
  Ville VARCHAR(100),  
  NomPro VARCHAR(100),  
  QteStock INT,  
  
  PRIMARY KEY(NPro, Ville)  
)  
  
CREATE TABLE Vente(  
  NomC VARCHAR(100),  
  DateV DATETIME,  
  QteV INT,  
  
  NPro INT,  
  
  PRIMARY KEY(NomC, DateV),  
  FOREIGN KEY(NPro) REFERENCES Produit  
)
```

Cette base présente des produits stockés dans diverses villes. Un tuple de produit est associé à chaque stock de produit. QteStock est la quantité du produit stockée dans une ville donnée. Les produits sont vendus à des clients. A chaque vente, un tuple contenant le nom du client NomC, la date DateV, le numéro de produit NPro et la quantité de produits QteV est enregistré dans la relation Vente.

Pour permettre l'estimation des performances des différents algorithmes de jointure présentés, les paramètres suivants sont fixés :

- a1 articles contenus dans p1 pages pour la relation produit
- a2 articles contenus dans p2 pages pour la relation vente
- b+2 pages de tampon qui représentent l'espace de travail disponible en mémoire pour les algos de jointure (Cet espace de travail doit aussi bien servir à la lecture qu'à l'écriture des pages).

Pour toutes les évaluations, on ne tiendra pas compte du cout des E/S lié à l'écriture du résultat car il est identique quel que soit l'algorithme de jointure choisi.

Application numérique :

$a1 = 1000; p1 = 100; a2 = 10000; p2 = 1000; b = 10$

**Question 1 :** Donner un exemple d'occurrences des relations produit et vente tel que la table produit possède 4 tuples, la table vente possède 5 tuples et la jointure en possède 7.

Produit	NPro	Ville	Vente	NomC	NPro
	1	Lille		A	1
	1	Valenciennes		B	2
	2	Metz		C	1
	3	Lille		B	2
				C	3

Jointure par boucle imbriquée :

Vente X Produit	NomC	#NPro	NPro	Ville
	A	1	1	L

Vente X Produit	NomC	#NPro	NPro	Ville
×2	A	1	1	V
	B	2	2	M
	C	1	1	L
	C	1	1	V
	C	3	3	L

On a bien 7 tuples.

## Question 2 : Produit et Vente sont modifiés pour la suite de la façon suivante :

```
CREATE TABLE Produit(
  NPro INT,
  Ville VARCHAR(100),
  NomPro VARCHAR(100),
  QteStock INT,

  !Changement de la clé primaire :
  PRIMARY KEY(NPro)
)

CREATE TABLE Vente(
  NomC VARCHAR(100),
  DateV DATETIME,
  QteV INT,

  NPro INT,

  PRIMARY KEY(NomC, DateV),
  FOREIGN KEY(NPro) REFERENCES Produit
)
```

Les 2 relations sont implantés sous forme de fichiers séquentiels. Comment réalisez-vous la jointure sur le numéro de produit ? Calculer le nombre d' $E/S$  nécessaires à l'exécution de l'algorithme proposé.

### Nombre d'entrées sorties :

Calculer un cout :

- *étape 1* : Mettre les b premières pages de produit puis la page 1 de vente.
- *étape 2* : Mettre les b premières pages de produit puis la page 2 de vente.
- .
- .
- .
- *étape n* : Mettre les b premières pages de produit puis la page n de vente.
- *étape n+1* : Mettre les b suivantes pages de produit puis la page 1 de vente.
- .
- .
- .

**Dans le pire des cas :**  $b = 1$ .

Cout en entrées sorties :  $C_{E/S}$

$C_{E/S} = p_2 + p_2 \times p_1 = 101000$  Si Vente comme relation externe.

$C_{E/S} = p_1 + p_1 \times p_2 = 100100$  Si Produit comme relation externe.

**Dans le meilleur des cas :**  $b \geq p_1$ .

$C_{E/S} = p_1 + p_2 = 1100$

Cas de l'énoncé :  $b = 10$ .

$$C_{E/S} = p_2 + \frac{p_1}{10} \times p_2 = 11000$$

---

**Question 3 :** Le système dispose d'un algorithme de tri capable de trier un fichier de  $p$  pages en  $2p \times \log_b(p) E/S$  ( $b$  = taille du tampon alloué à l'algorithme de tri fusion utilisé). Donner le nombre d' $E/S$  nécessaires.

**Jointure par tri fusion :**

- étape 1 : Tri des 2 relations sur l'attribut de jointure.
- étape 2 : fusion.

$$C_{E/S}(JTF) = C_{E/S}(Tri) + C_{E/S}(Fusion)$$

Produit	NPro	Vente	NPro
	1		1
	2		1
	3		1
	4		1
			2
			2
			2
			.
			.
			.

$$C_{E/S}(JTF) = C_{E/S}(Tri(Produit)) + C_{E/S}(Tri(Vente)) + C_{E/S}(Fusion)$$

$$C_{E/S}(Fusion) = p_1 + p_2 = 1100$$

$$\Leftrightarrow C_{E/S}(JTF) = 2p_1 \times \log_b(p_1) + 2p_2 \times \log_b(p_2) + 1100 = 7500 E/S$$

---

**Question 4 :** Les relations produit et vente sont implantées sous la forme de fichiers aléatoires hachés sur NPro. Donner le nombre d' $E/S$  nécessaires.

**Jointure par hachage :**

- étape 1 : hachage (on récupère les tuples de façon homogène dans différentes partitions)
- étape 2 : fusion.

$$C_{E/S}(JH) = C_{E/S}(Hachage) + C_{E/S}(Fusion)$$

$$\Leftrightarrow C_{E/S}(JH) = 2p_1 + 2p_2 + p_1 + p_2 = 3300 E/S$$