



Bases de Données Avancées

TD : Bases de données distribuées -Fragmentation-

- USTHB Master 01 IL -
M. AZZOUZ

Dernière mise à jour : Mai 2020

Exercice 01

➤ L'administrateur du portail web d'une entreprise spécialisée en vente par internet (e-commerce) dispose d'une table *Utilisateur* dont le schéma est le suivant :

Utilisateur (IDU, Nom, Prénom, email, mot_de_passe, Date_Enregistrement, Date_Dernier_Accès, Ville, Pays, Code_Postal, Téléphone, Age).

Soient les requêtes R1, R2 et R3 suivantes :

-R1: Select Nom, Prénom, email From Utilisateur where Date_Dernier_Accès < '01-03-2011';

-R2: Select Ville, Pays, Age From Utilisateur where Date_Enregistrement < '01-01-2011';

- R3: Select Nom, Prénom, Ville From Utilisateur where Age>18;

➤ **Objectifs de la fragmentation:** Pour optimiser les requêtes R1 et R2, l'administrateur décide de fragmenter verticalement la table *Utilisateur* en **trois fragments** : *User1*, *User2* et *User3*. Ces fragments sont définis pour que chaque requête **charge uniquement les colonnes nécessaires à son exécution** (R1 charge *USER1*, R2 Charge *USER2*).

Exercice 01

1. Donnez les expressions algébriques permettant de représenter chaque fragment
 - a. Fragment USER1:

R1:

Select **Nom, Prénom, email**

From Utilisateur

where **Date_Dernier_Accès** < '01-03-2011';



USER1 = $\Pi_{IDU, \text{Nom}, \text{Prénom}, \text{Email}, \text{Date_Dernier_Accès}}$ (Utilisateur)

Exercice 01

1. Donnez les expressions algébriques permettant de représenter chaque fragment
- b. Fragment USER2:

R2:

Select **Ville, Pays, Age**

From Utilisateur

where **Date_Enregistrement** < '01-01-2011';



$USER2 = \Pi_{IDU, Ville, Pays, Age, Date_Enregistrement} (Utilisateur)$

Exercice 01

1. Donnez les expressions algébriques permettant de représenter chaque fragment
- c. Fragment USER3:

- **USER1** (IDU, Nom, Prénom, Email, Date_Dernie_Accès).
- **USER2** (IDU, Ville, Pays, Age, Date_Enregistrement).
- **USER3** → le reste de colonnes de la relation **Utilisateur** (IDU, Nom, Prénom, email, **mot_de_passe**, Date_Enregistrement, Date_Dernier_Accès, Ville, Pays, **Code_Postal**, **Téléphone**, Age)



USER3 = Π IDU, Mot_de_passe, Code_Postal, Téléphone (Utilisateur)

Exercice 01

2. Donnez la réécriture des requêtes R1 et R2 sur la table Utilisateur fragmentée.

La réécriture de la requête passe par un processus d'optimisation suivant:

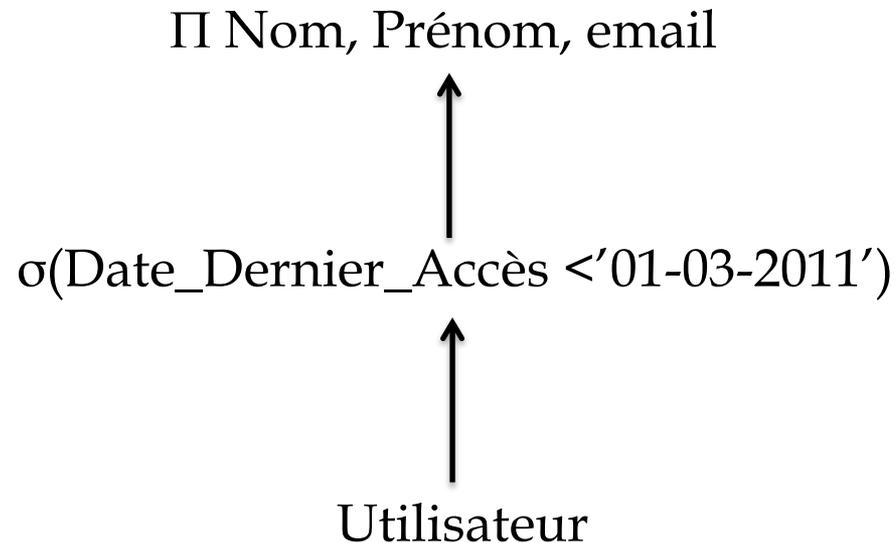
- Dessiner l'arbre global (en considérant la relation avant fragmentation).
- Dessiner l'arbre canonique: remplacer la relation globale par la jointure des fragments pour une fragmentation verticale ou bien l'union dans le cas de la fragmentation horizontale.
- Utiliser les règles générales d'optimisation (faire descendre les restrictions, les projection et remonter les jointures).
- Eliminer les fragments vides (« un fragment vide dans le cas de la fragmentation verticale est le fragment dont le résultat de la projection finale est que l'attribut **clé primaire** », « un fragment est vide lorsque la **condition de la restriction est contradictoire avec la condition de création de fragment** »).

Exercice 01

2. Donnez la réécriture des requêtes R1 et R2 sur la table Utilisateur fragmentée.

a. La réécriture de la requête R1:

➤ Arbre global:

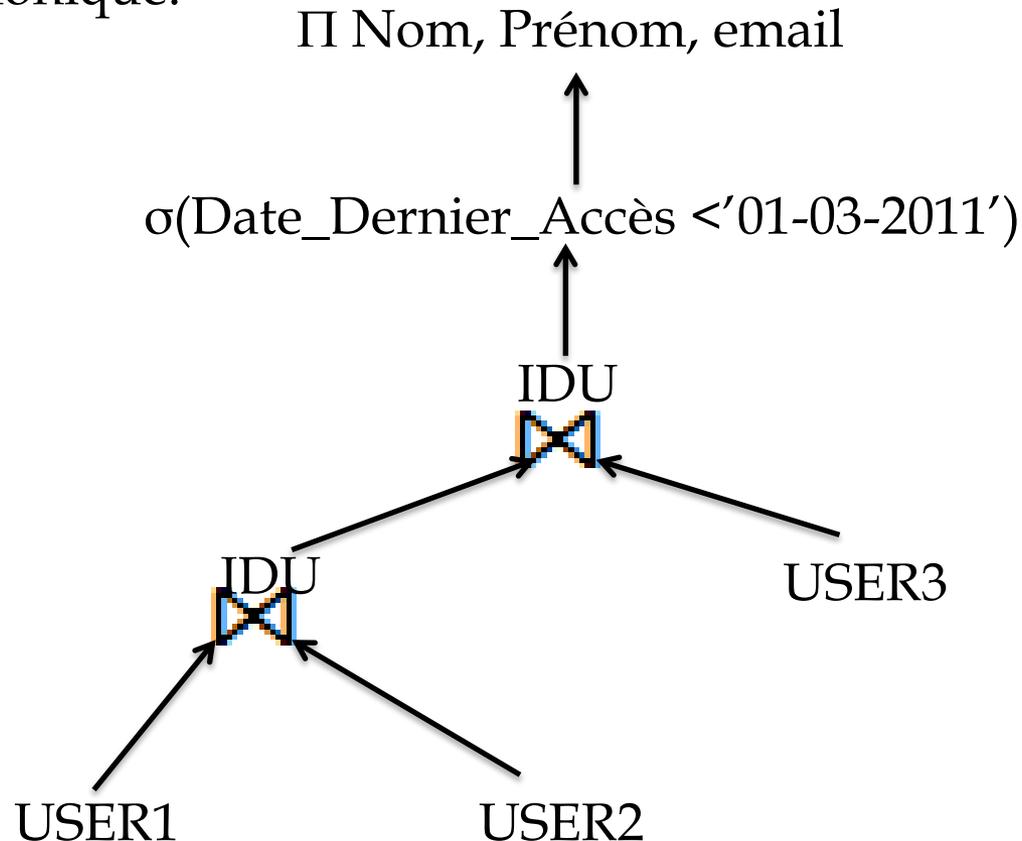


Exercice 01

2. Donnez la réécriture des requêtes R1 et R2 sur la table Utilisateur fragmentée.

a. La réécriture de la requête R1:

➤ Arbre canonique:



Exercice 01

2. Donnez la réécriture des requêtes R1 et R2 sur la table Utilisateur fragmentée.

a. La réécriture de la requête R1:

➤ Règles d'optimisation générales:

Π Nom, Prénom, email

IDU

Π IDU, Nom, Prénom, email

Π IDU

IDU

USER3

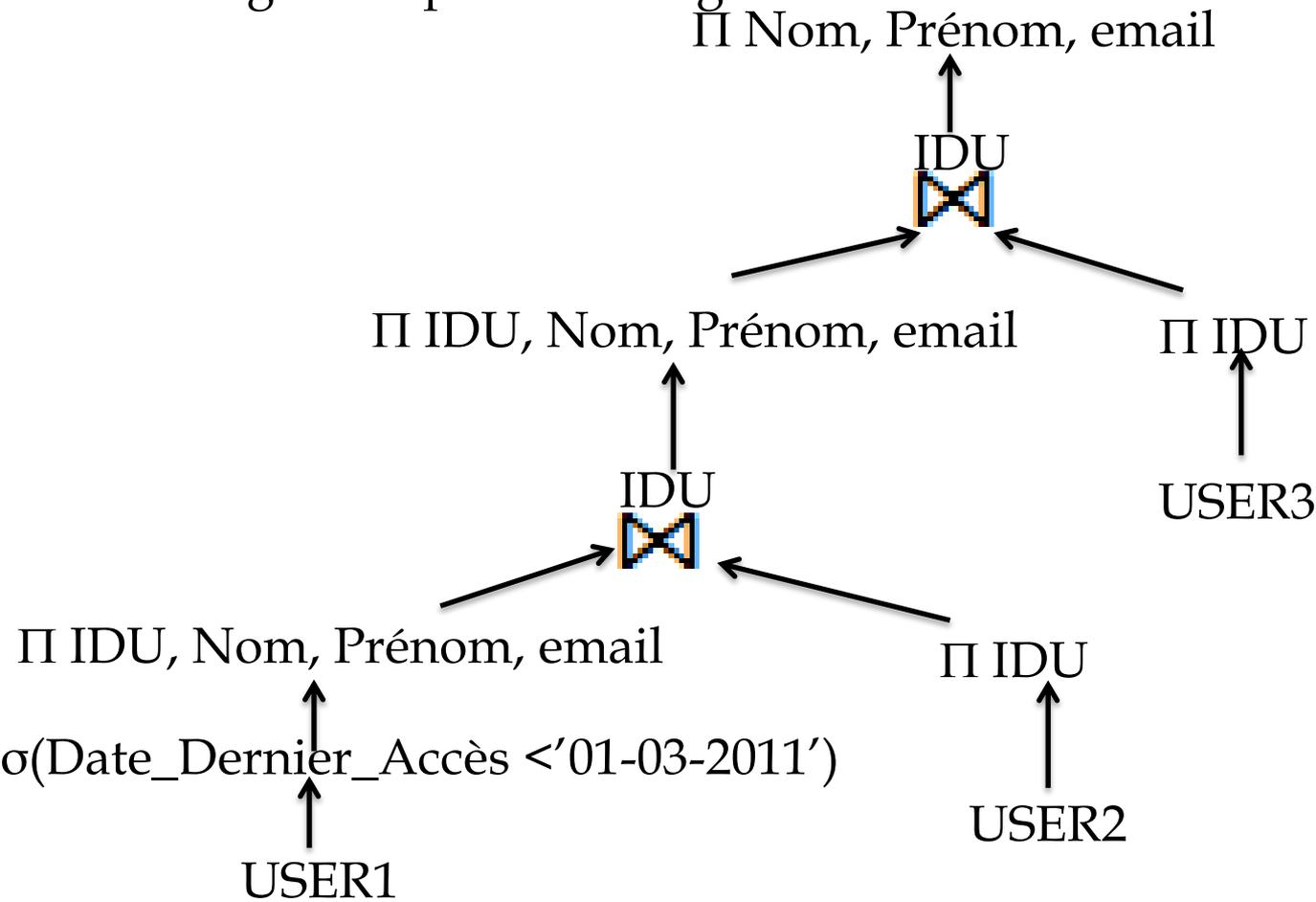
Π IDU, Nom, Prénom, email

Π IDU

$\sigma(\text{Date_Dernier_Accès} < '01-03-2011')$

USER2

USER1

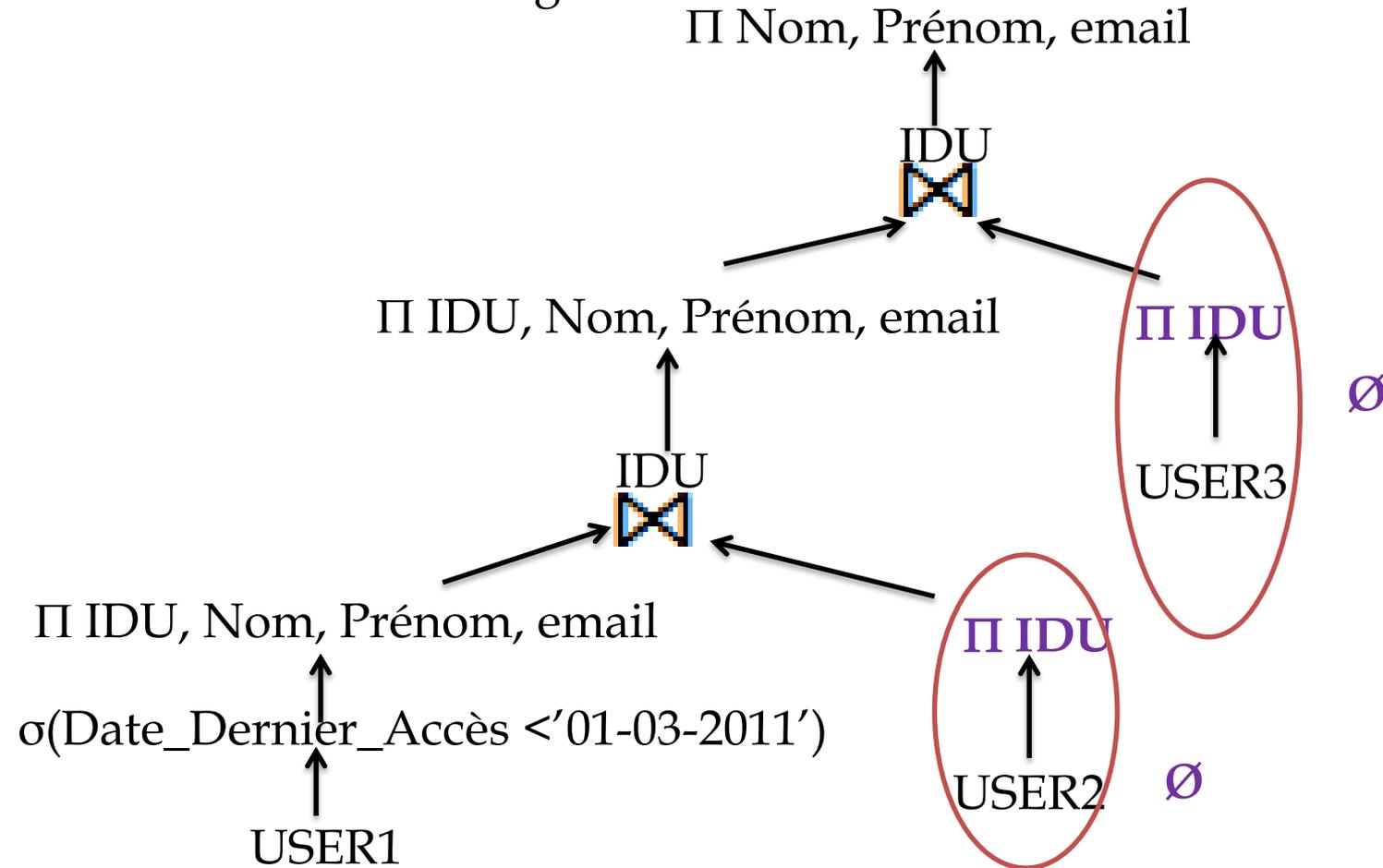


Exercice 01

2. Donnez la réécriture des requêtes R1 et R2 sur la table Utilisateur fragmentée.

a. La réécriture de la requête R1:

➤ Eliminer les fragments vides:

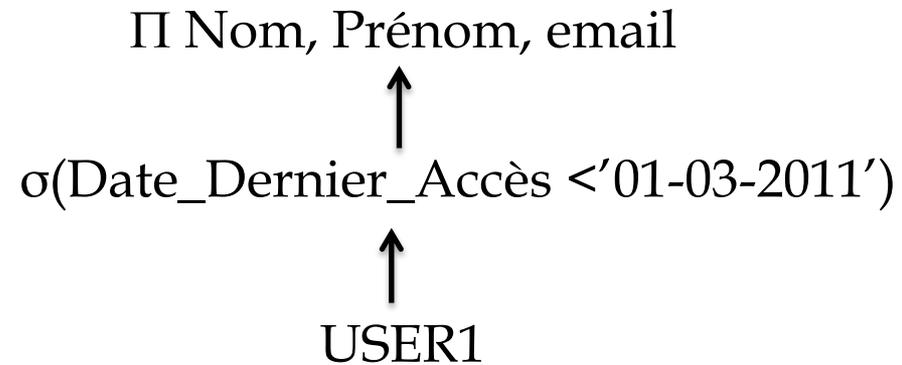
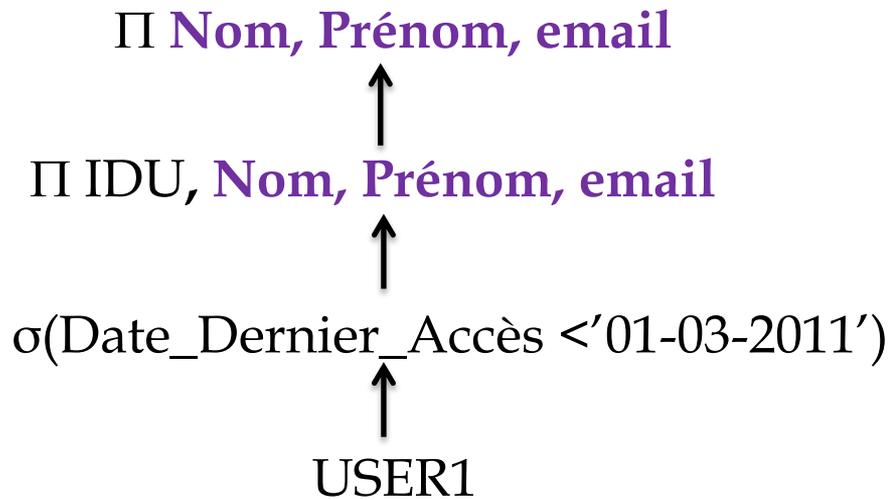


Exercice 01

2. Donnez la réécriture des requêtes R1 et R2 sur la table Utilisateur fragmentée.

a. La réécriture de la requête R1:

➤ Arbre simplifié



Exercice 01

2. Donnez la réécriture des requêtes R1 et R2 sur la table Utilisateur fragmentée.

a. La réécriture de la requête R1:

➤ R1 réécrite:

Π Nom, Prénom, email
↑
 σ (Date_Dernier_Accès < '01-03-2011')
↑
USER1

```
Select Nom, Prénom, email  
From USER1  
where Date_Dernier_Accès < '01-03-2011';
```

Exercice 01

2. **Donnez la réécriture des requêtes R1 et R2 sur la table Utilisateur fragmentée.**
 - b. La réécriture de la requête R2:
 - En utilisant le même processus d'optimisation on aura R2 réécrite:

```
Select Ville, Pays, Age  
From USER2  
where Date_Enregistrement < '01-01-2011';
```

Exercice 01

3. Donnez une réécriture de la requête R3 sur la table Utilisateur fragmentée. Que concluez-vous ?

➤ Arbre global:

Π Nom, Prénom, Ville

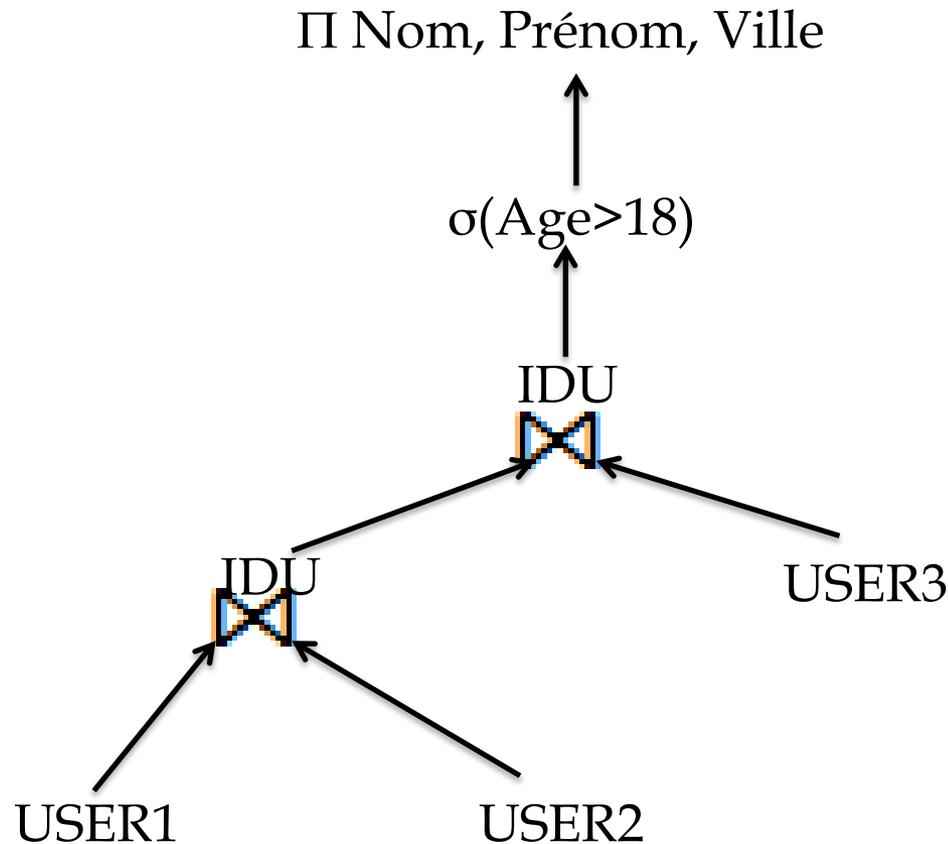
$\sigma(\text{Age} > 18)$

Utilisateur

Exercice 01

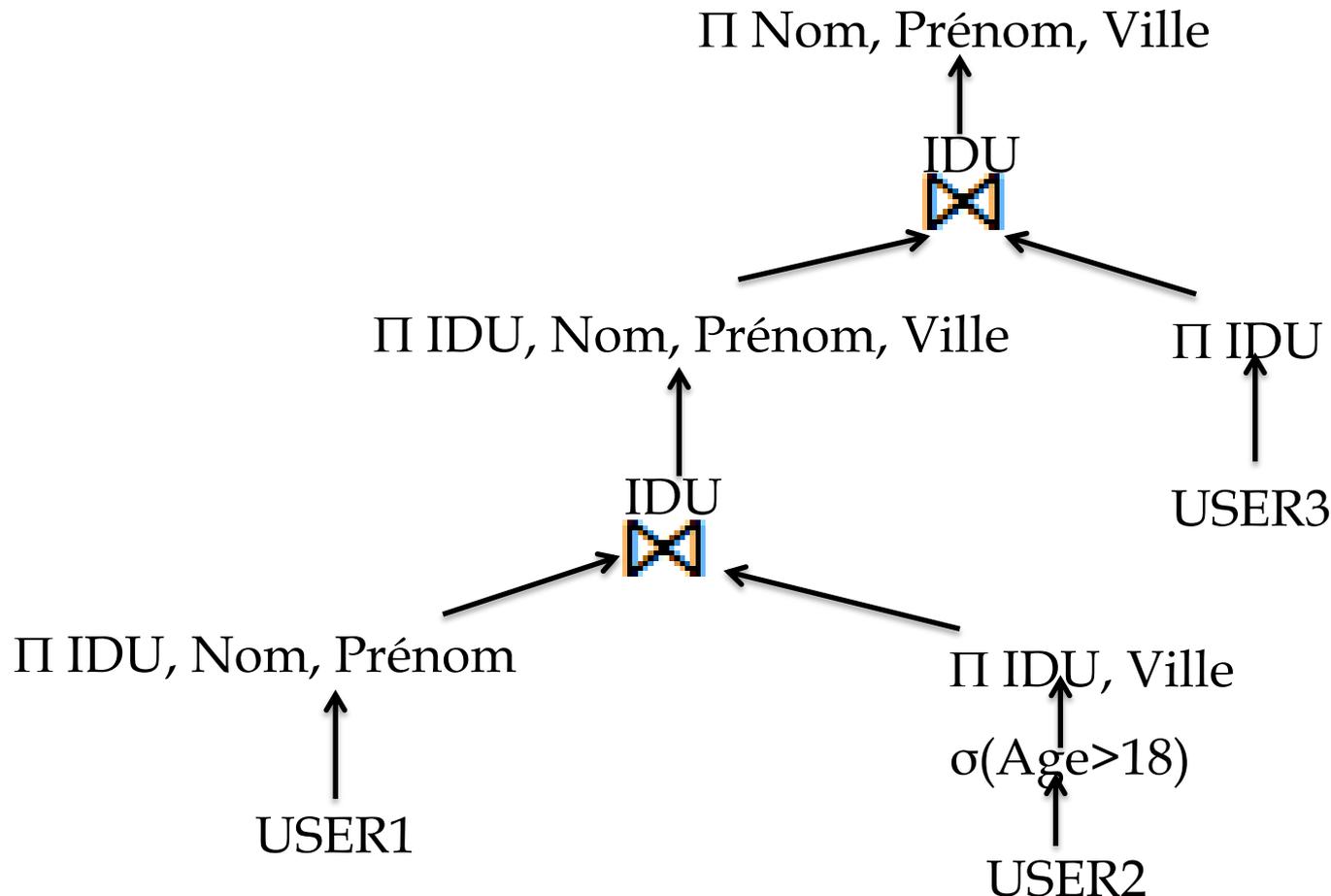
3. Donnez une réécriture de la requête R3 sur la table Utilisateur fragmentée. Que concluez-vous ?

➤ Arbre canonique:



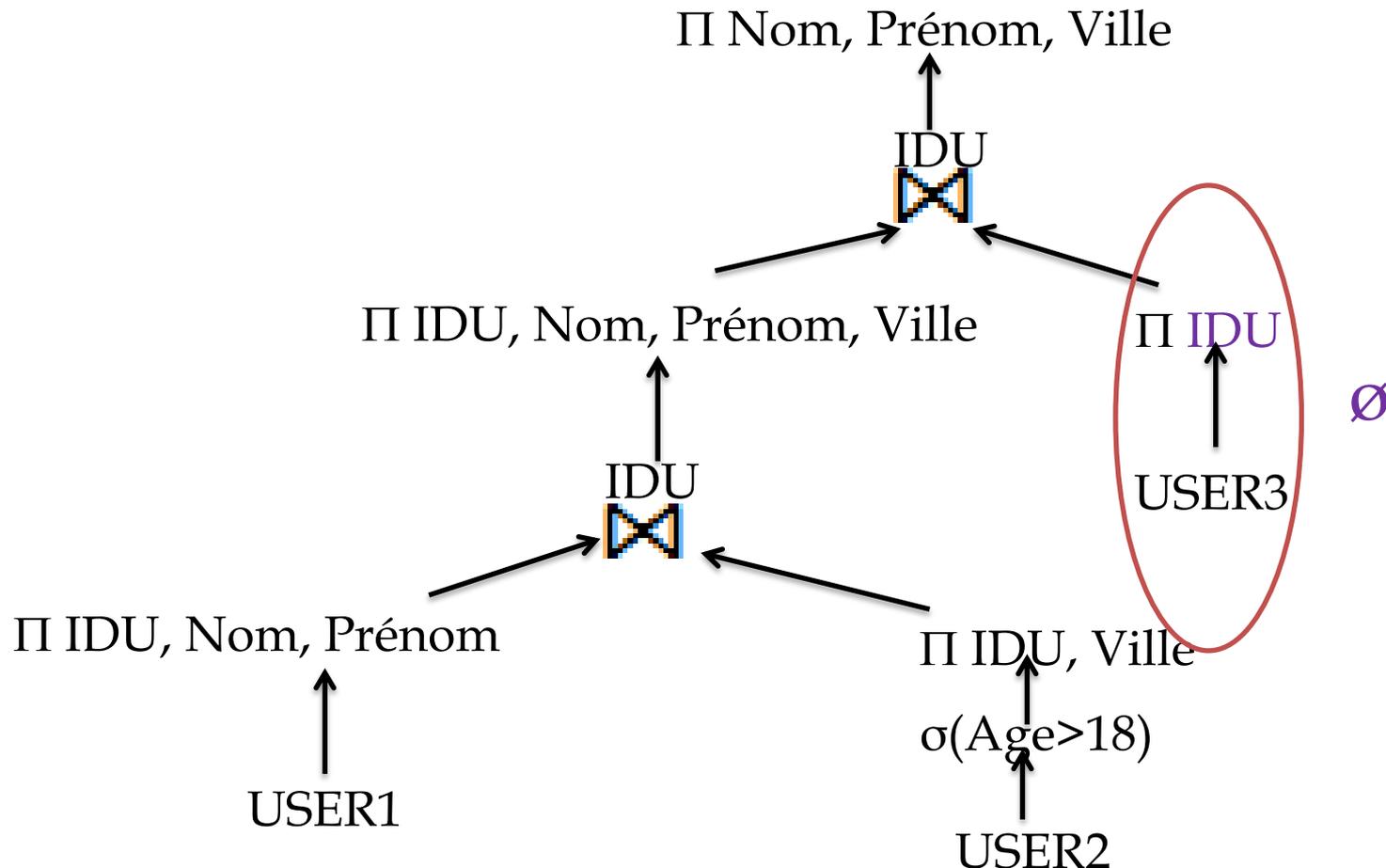
Exercice 01

3. Donnez une réécriture de la requête R3 sur la table Utilisateur fragmentée. Que concluez-vous ?
- Règles d'optimisation générales:



Exercice 01

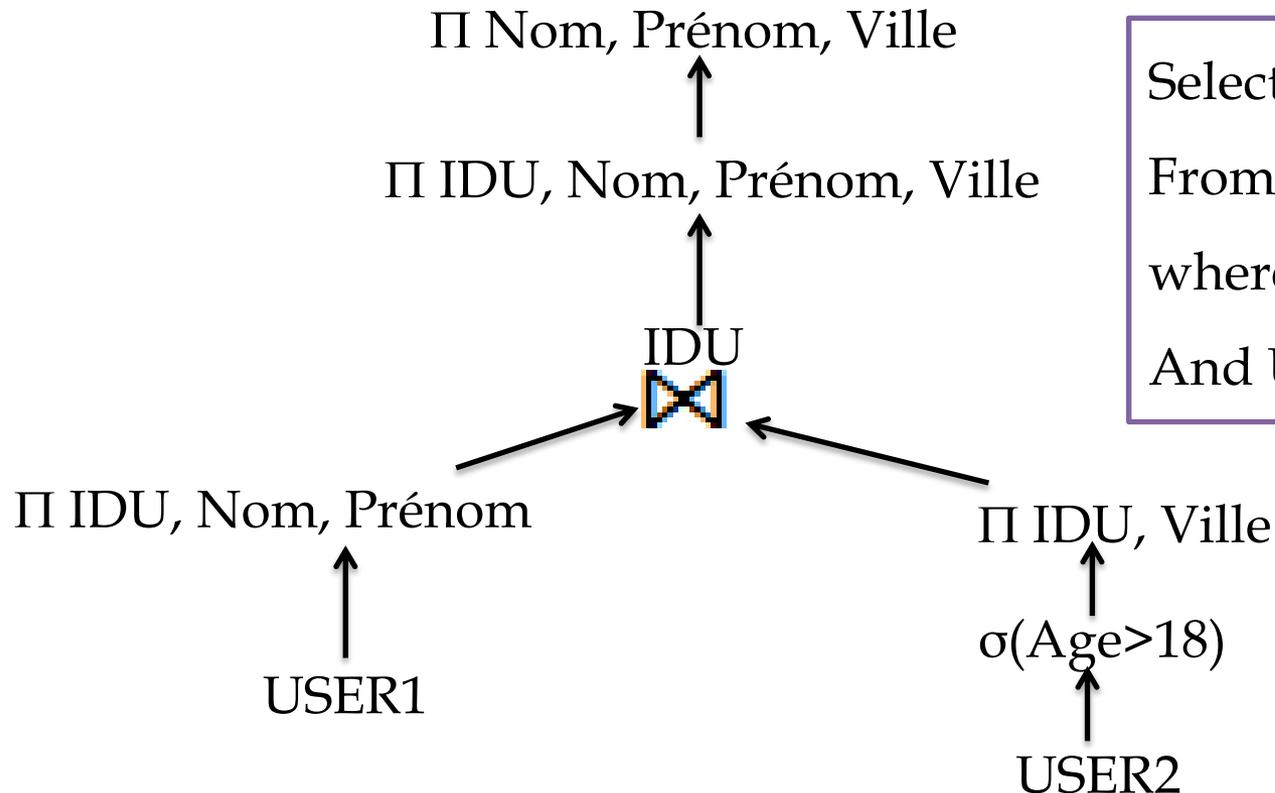
3. Donnez une réécriture de la requête R3 sur la table Utilisateur fragmentée. Que concluez-vous ?
- Eliminer les fragments vides:



Exercice 01

3. Donnez une réécriture de la requête R3 sur la table Utilisateur fragmentée. Que concluez-vous ?

➤ Arbre simplifié:



```
Select Nom, Prénom, Ville
From USER1 U1, USER2 U2
where U2.Age>18
And U1.IDU=U2.IDU;
```

Exercice 01

- 3. Donnez une réécriture de la requête R3 sur la table Utilisateur fragmentée. Que concluez-vous ?**

➤ **Conclusion:**

Les deux requêtes R1 et R2 sont réécrites chacune sur un fragment car tous les attributs référencés par chacune des requêtes appartiennent au même fragment. La requête R3 accède à deux fragments, alors une opération de jointure supplémentaire entre USER1 et USER2 est ajoutée à la requête pour récupérer tous les attributs référencés.

Exercice 01

L'entreprise est répartie géographiquement en trois sites distants : Alger, Oran et Constantine. Une allocation simple de ces fragments consiste à mettre chacun sur un site : User1 à Alger, User2 à Oran et User3 à Constantine. Supposons que chaque attribut est codé sur 50 octets, que la table Utilisateur contient 1000 000 de tuples et que la taille d'une page système est de 6000 octets. Le nombre d'utilisateurs enregistrés avant le 01-01-2011 est de 50 000, le nombre d'utilisateurs dont le dernier accès a été fait avant le 01-03-2011 est de 5000, le nombre d'utilisateurs âgés de plus de 18 ans est de 600 000.

Exercice 01

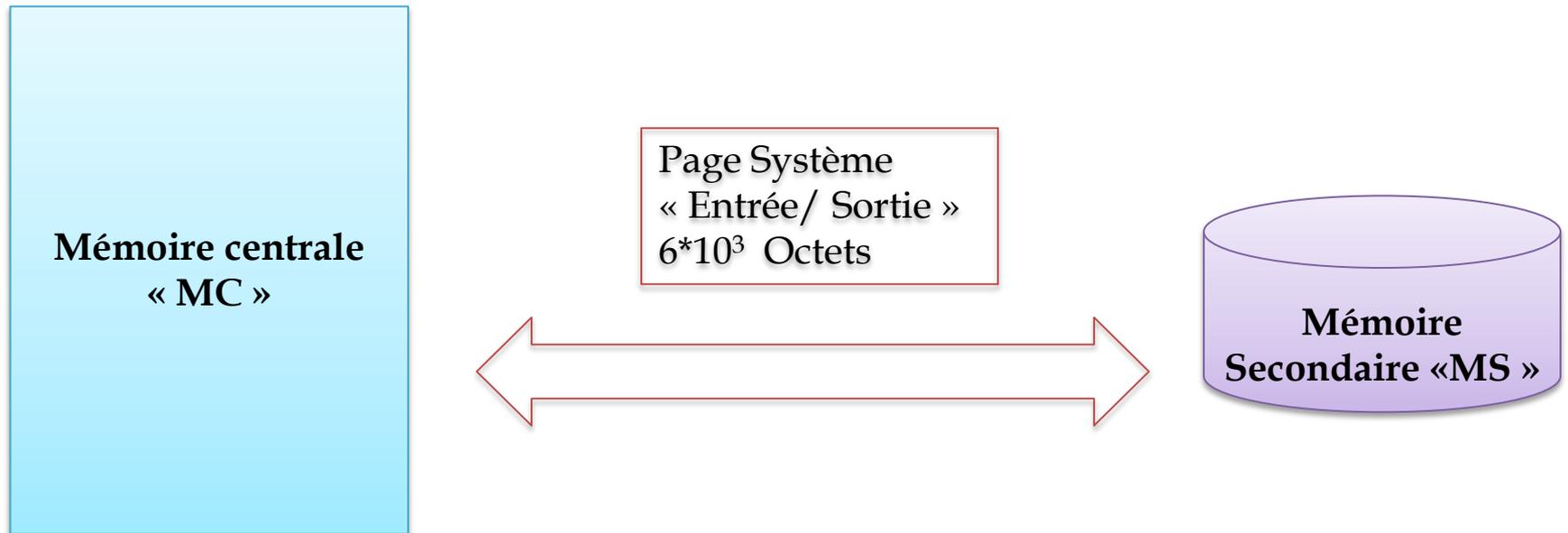
Dictionnaire				
Relation	Utilisateur	USER1	USER2	USER3
Degré	12 attributs	5 attributs	5 attributs	4 attributs
Taille attribut	50 octets	50 octets	50 octets	50 octets
Taille tuple	12*50 octets	5*50 octets	5*50 octets	4*50 octets
Cardinalité	10^6 tuples	10^6 tuples	10^6 tuples	10^6 tuples

Exercice 01

Statistiques	
Description	Valeurs
Le nombre de clients enregistrés avant le 01-01-2011	$5 \cdot 10^4$ tuples
le nombre de clients dont le dernier accès a été fait avant le 01-03-2011	$5 \cdot 10^3$ tuples
le nombre de clients âgés de plus de 18 ans	$8 \cdot 10^5$ tuples

Exercice 01

Caractéristiques matériels:



Exercice 01

Coût d'exécution dans le cas centralisé (avant fragmentation) est calculé par la formule suivante:

- Coût d'exécution = Coût de chargement + Coût CPU
- Coût CPU est négligé dans cet exercice
- Coût de chargement = nombre de page à charger * temps de chargement d'une page.
- Nombre de page à charger = taille à charger / taille d'une page système.
- Taille à charger = nombre de tuples * taille d'un tuple.
- Taille d'un tuple = somme des tailles des attributs.

Exercice 01

4. Calculez le coût d'exécution de R1, R2 et R3 avant fragmentation. Ce coût est exprimé en nombre de pages chargées de la mémoire secondaire pour exécuter chaque requête. Exprimez ce coût en secondes sachant que le temps de chargement d'une page est estimé à 1 milliseconde.

$$\text{Coût d'exécution} = ((10^6 * (12 * 50)) / 6 * 10^3) * 10^{-3}$$

$$\text{Coût d'exécution} = 100 \text{ secondes}$$

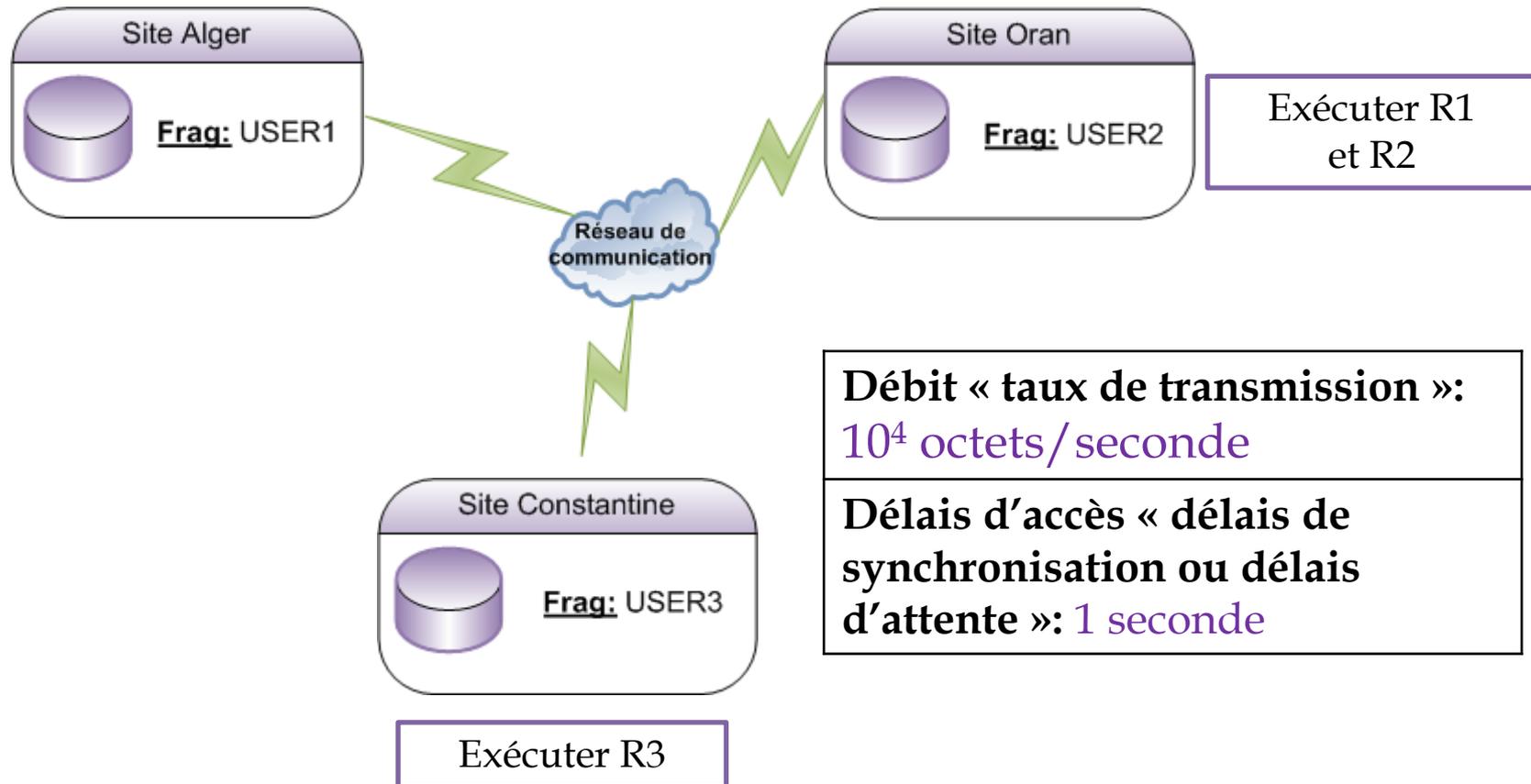
Exercice 01

5. Sachant que R1 et R2 ont été lancées à Oran et R3 à Constantine.

- **Donnez les stratégies d'exécution possibles pour R1 et R2.**
- **Calculez le coût d'exécution global de R1 et R2 dans chaque stratégie. Rappelons que ce coût d'exécution regroupe le coût de chargement de données et le coût de communication. On suppose que le taux de transmission est de 10000 octets/seconde et que le délai d'attente avant de mettre les données sur le canal de transmission est de 1 seconde.**
- **Que concluez-vous ?**
- **Donnez toutes les stratégies d'exécution de la requête R3**

Exercice 01

Caractéristiques du réseau



Exercice 01

❑ Requête R2:

➤ La requête R2 est réécrite sur USER2 et lancée à partir de site d'Oran. Le fragment USER2 est alloué au site d'Oran. La requête R2 est une requête locale pas de transmission de données d'où on a une seule stratégie.

➤ Coût d'exécution=Coût de chargement de fragment USER2

➤ Coût d'exécution= $((10^6 * (5 * 50)) / 6 * 10^3) * 10^{-3}$

➤ Coût d'exécution=41.66 secondes

Exercice 01

❑ Requête R1:

➤ La requête R1 est réécrite sur USER1 et lancée à partir de site d'Oran. Le fragment USER1 est alloué au site d'Alger. La requête R1 est une requête globale car les données sont distantes. Deux stratégies sont possibles.

▪ **Stratégie 01:** Exécuter R1 au site d'Alger et envoyer les résultats au site lançant la requête le site d'Oran.

▪ **Stratégie 02:** Transmettre le fragment USER1 au site d'Oran et exécuter la requête R1 localement.

➤ Coût d'exécution dans le cas distribué est donné par la formule suivante:

▪ $\text{Coût d'exécution} = \text{Coût de chargement} + \text{Coût CPU} + \text{Coût de communication de données.}$

▪ $\text{Coût de communication de données} = \text{délais d'attente} + \text{taille à transférer} / \text{débit de transmission.}$

Exercice 01

❑ **Requête R1:** Select **Nom, Prénom, email** From USER1 where Date_Dernier_Accès < '01-03-2011';

■ **Stratégie 01:** Exécuter R1 au site d'Alger et envoyer les résultats au site lançant la requête le site d'Oran.

■ Coût d'exécution= Coût de chargement+ Coût CPU+ Coût de communication de données.

■ Coût de chargement= $((10^6 * (5 * 50)) / 6 * 10^3) * 10^{-3} = 41.66$ secondes

■ D'après les statistiques **le nombre de clients dont le dernier accès a été fait avant le 01-03-2011 est égale à $5 * 10^3$ tuples .**

■ Coût de communication de données = délais d'attente+ taille à transférer/débit de transmission.

■ Coût de communication de données = $1 + ((5 * 10^3 * (3 * 50)) / 10^4) = 75$ secondes.

■ **Coût d'exécution=41.66+75=116.66 secondes.**

Exercice 01

❑ **Requête R1:** Select **Nom, Prénom, email** From USER1 where Date_Dernier_Accès < '01-03-2011';

▪ **Stratégie 02:** Transmettre le fragment USER1 au site d'Oran et exécuter la requête R1 localement.

▪ Coût d'exécution= Coût de chargement+ Coût CPU+ Coût de communication de données.

▪ Coût de communication de données = délais d'attente+ (taille à transférer/débit de transmission).

▪ Coût de communication de données = $1 + ((10^6 * (5 * 50)) / 10^4)$.

▪ Coût de communication de données = 25001 seconde = 6.94 heures.

▪ Coût de chargement = $((10^6 * (5 * 50)) / 6 * 10^3) * 10^{-3} = 41.66$ secondes

▪ **Coût d'exécution = 25001 + 41.66 = 25 042,66 secondes = 6.95 heures.**

Exercice 01

□ **Donnez toutes les stratégies d'exécution de la requête R3.**

R3 est lancé à partir de site Constantine et nécessite la jointure de deux fragments USER1 (à Alger) et USER2(à Oran)

Stratégie	Description
Stratégie 1	Envoyer USER 1 à Oran, faire la jointure avec USER 2, envoyer le résultat à Constantine
Stratégie 2	Envoyer USER 2 à Alger, faire la jointure avec USER 1, envoyer le résultat à Constantine
Stratégie 3	Envoyer USER 1 et USER 2 à Constantine et exécuter la requête à Constantine
Stratégie 4	Exécuter la restriction Age>18 à Oran, envoyer le résultat à Alger, faire la jointure à Alger ensuite envoyer le résultat à Constantine.

Exercice 01

6. Supposons qu'une table T composée de n attributs est fragmentée en m partitions ($m < n$) où chaque partition T_i contient n_i attributs. Donnez l'algorithme de réécriture d'une requête quelconque sur la table T fragmentée.

Exercice 01

Algorithme de Réécriture

Entrées : m partition T_1, T_2, \dots, T_m , Une requête R

Sortie : R Réécrite

Début

Soit $A = \{A_1, A_2, \dots, A_k\}$, l'ensemble des attributs de T référencés par R (sauf la clé).

Soit A_i l'ensemble des attributs de chaque fragment T_i (sauf la clé)

Pour $i=1$ à m

Faire

Si A inclus ou égal A_i **alors** Le fragment T_i est valide

Remplacer T dans la clause FROM par T_i

Sortir de l'algorithme

Sinon Si A intersection $A_i \neq$ ensemble vide **Alors** T_i contient quelques attributs référencés par R Ajouter T_i dans la liste des fragments valides

Sinon T_i ne contient aucun attribut de R, Il n'est pas valide. Ne rien faire

Fin Faire

Remplacer T dans la clause FROM par des jointures entre tous les fragments valides.

Fin

Exercice 01

7. La fragmentation verticale n'est pas supportée sous Oracle. Proposez une implémentation de la fragmentation verticale en utilisant les vues.

□ Nous pouvons utiliser les Vues pour l'implémenter comme suit :

■ Créer une vue pour chaque fragment vertical. Si un fragment T_i contient les attributs A_1, A_2, \dots, A_l de T alors la vue sera créée comme suit :

Create View T_i AS Select A_1, A_2, \dots, A_l FROM T

■ Effectuer la réécriture des requêtes sur les vues : nous pouvons utiliser pour cela l'algorithme de réécriture donné dans la question précédente.

Exercice 02

❑ Soit le schéma BD suivant :

PRODUCTION (NUMSERIE, COMPOSANT, MODELE, QUANTITE, MACHINE)

VENTE (NUMSERIE, LOT, NOMCLIENT, NOMVENDEUR, MONTANT)

CLIENT (NOMCLIENT, RUECLIENT, VILLECLIENT, PAYSCLIENT)

VENDEUR (NOMVENDEUR, RUEVENDEUR, VILLEVENDEUR, PAYSVENDEUR)

❑ L'entreprise possède quatre sites de production, tel que chaque site est responsable de la production d'un composant : Unité Centrale, clavier, écran et câble produits respectivement par les sites Tarf, Sétif, Annaba et Alger.

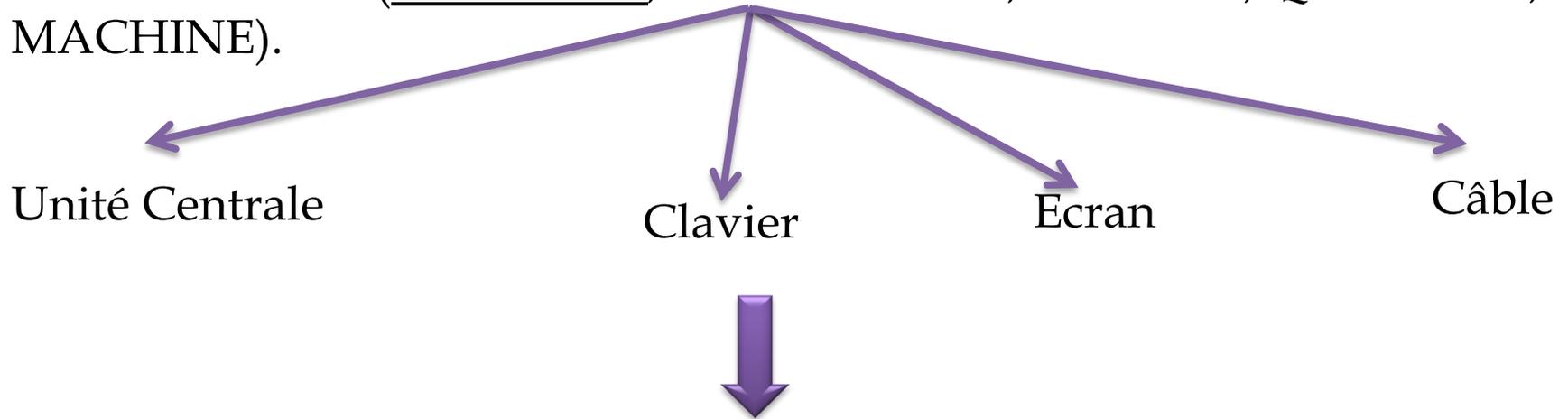
❑ Les points de vente de l'entreprise sont à Sétif, Annaba et Alger. A l'exception des clients de Tarf qui s'adressent au point de vente de Annaba, tous les autres clients s'adressent aux points de vente existant de leurs villes respectives.

Exercice 02

1. Proposer une fragmentation des quatre relations Production, Vente, client et Vendeur, en justifiant vos choix et précisant le type de fragmentation utilisé.

➤ **Table production:**

PRODUCTION (NUMSERIE, COMPOSANT, MODELE, QUANTITE, MACHINE).



Fragmentation Horizontale Primaire « FHP » de la table production selon l'attribut composant : Unité Centrale, clavier, écran et câble.

Exercice 02

➤ FHP de la table production:

Fragment	Expression Algébrique
Prod_UC	$\text{Prod_UC} = \sigma_{\text{COMPOSANT} = \text{"Unité Centrale"}}(\text{Production})$
Prod_Clavier	$\text{Prod_Clavier} = \sigma_{\text{COMPOSANT} = \text{"Clavier"}}(\text{Production})$
Prod_Ecran	$\text{Prod_Ecran} = \sigma_{\text{COMPOSANT} = \text{"Ecran"}}(\text{Production})$
Prod_Câble	$\text{Prod_Câble} = \sigma_{\text{COMPOSANT} = \text{"Câble"}}(\text{Production})$

Exercice 02

➤ Table Vente:

VENTE (NUMSERIE*, LOT, NOMCLIENT, NOMVENDEUR, MONTANT)

Vente est fragmentée selon une fragmentation horizontale dérivée « **FHD** » de la table production donc on regroupera les ventes par rapport aux composants produits;

Fragment	Expression Algébrique
Vente_UC	Vente_UC = Vente \bowtie Prod_UC
Vente_Clavier	Vente_Clavier = Vente \bowtie Prod_Clavier
Vente_Ecran	Vente_Ecran = Vente \bowtie Prod_Ecran
Vente_Câble	Vente_Câble = Vente \bowtie Prod_Câble

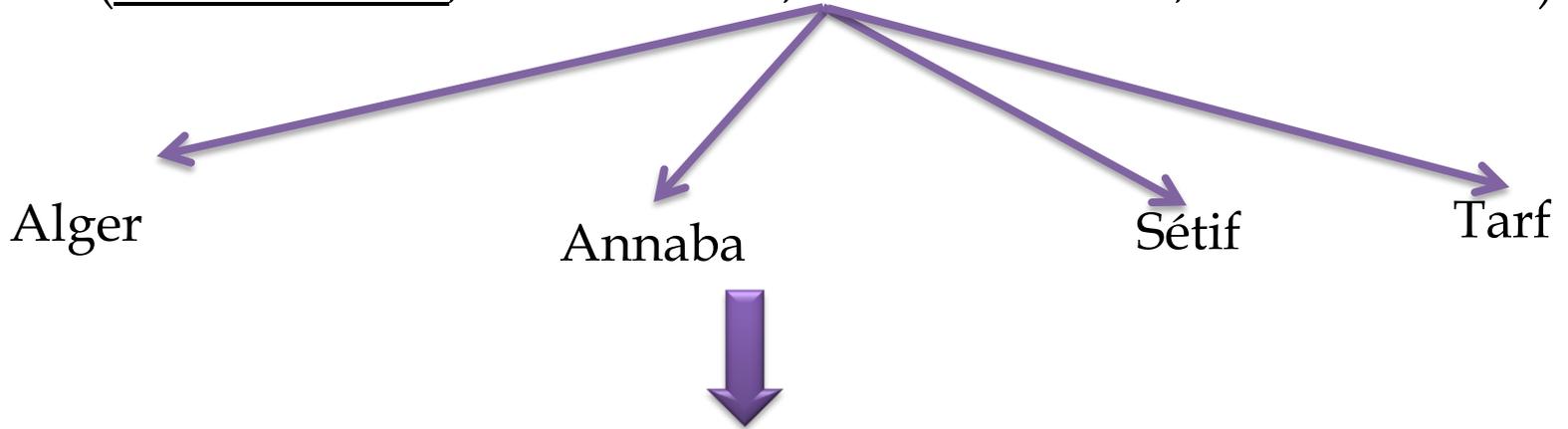
Semi-jointure:

Vente \bowtie Prod_Câble = Vente \bowtie (Π_{Numserie} (Prod_Câble))

Exercice 02

➤ **Table client:**

CLIENT (NOMCLIENT, RUECLIENT, **VILLECLIENT**, PAYSCLIENT)



Fragmentation Horizontale primaire « FHP » de la table CLIENT selon l'attribut VILLECLIENT : Sétif, Alger, Annaba et Tarf.

Exercice 02

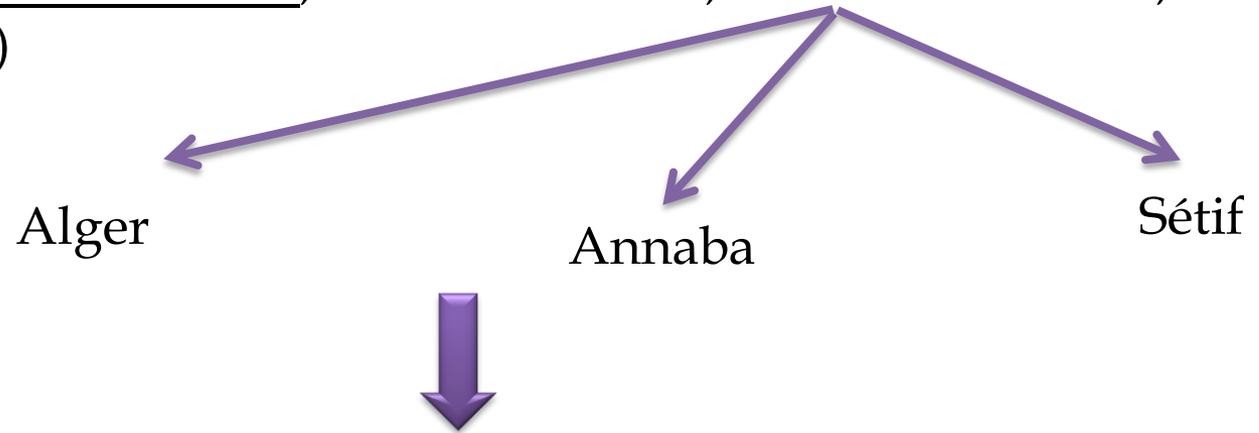
➤ FHP de la table client:

Fragment	Expression Algébrique
Client_ Sétif	$\text{Client_Sétif} = \sigma_{\text{VILLECLIENT} = \text{"Sétif"}}(\text{Client})$
Client_ Alger	$\text{Client_Alger} = \sigma_{\text{VILLECLIENT} = \text{"Alger"}}(\text{Client})$
Client_ Annaba_ Tarf	$\text{Client_Annaba_Tarf} = \sigma_{\text{VILLECLIENT} = \text{"Annaba"} \vee \text{VILLECLIENT} = \text{"Tarf"}}(\text{Client})$

Exercice 02

➤ **Table Vendeur:**

VENDEUR (NOMVENDEUR, RUEVENDEUR, VILLEVENDEUR, PAYSVENDEUR)



Fragmentation Horizontale primaire « FHP » de la table VENDEUR selon l'attribut VILLEVENDEUR : Sétif, Alger et Annaba.

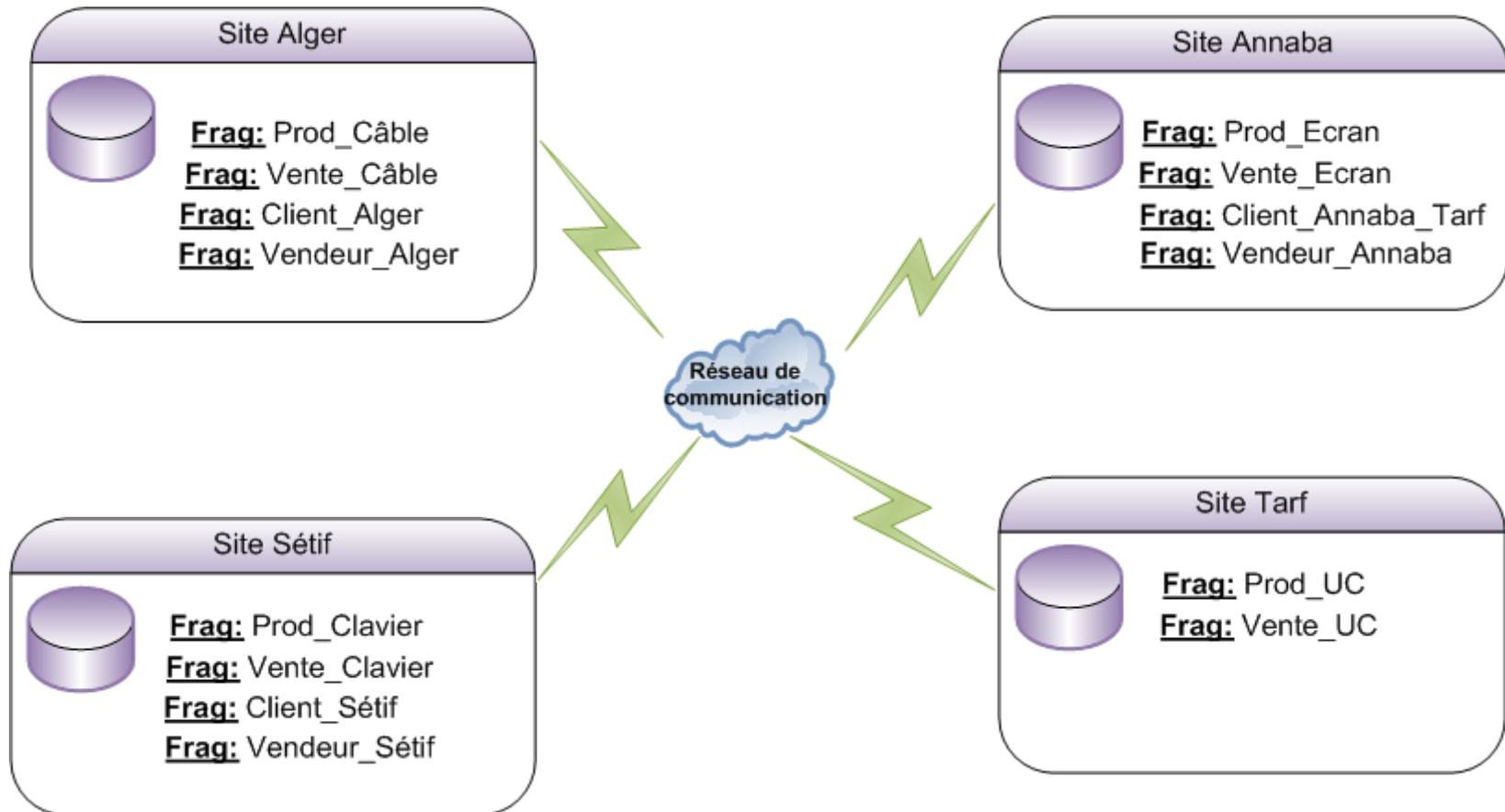
Exercice 02

➤ FHP de la table **Vendeur**:

Fragment	Expression Algébrique
Vendeur_Sétif	$\text{Vendeur_Sétif} = \sigma_{\text{VILLEVENDEUR} = \text{"Sétif"}}(\text{Vendeur})$
Vendeur_Alger	$\text{Vendeur_Alger} = \sigma_{\text{VILLEVENDEUR} = \text{"Alger"}}(\text{Vendeur})$
Vendeur_Annaba	$\text{Vendeur_Annaba} = \sigma_{\text{VILLEVENDEUR} = \text{"Annaba"}}(\text{Vendeur})$

Exercice 02

2. Proposer un schéma d'allocation de tous les fragments.



Exercice 02

3. Sachant que la fragmentation est transparente à l'utilisateur et que les requêtes suivantes sont émises à partir du site de Annaba:

- **R1** : Quelle est la quantité disponible du produit de numéro de série '77y6878'.
 - **R2** : La somme des quantités de produits par composant et modèle.
- a. Pour chaque requête donner le script SQL associé, l'arbre canonique ainsi que l'arbre algébrique simplifié.
 - b. Donner les différentes stratégies d'exécution de la requête R2.

Exercice 02

R1 : Quelle est la quantité disponible du produit de numéro de série '77y6878'.

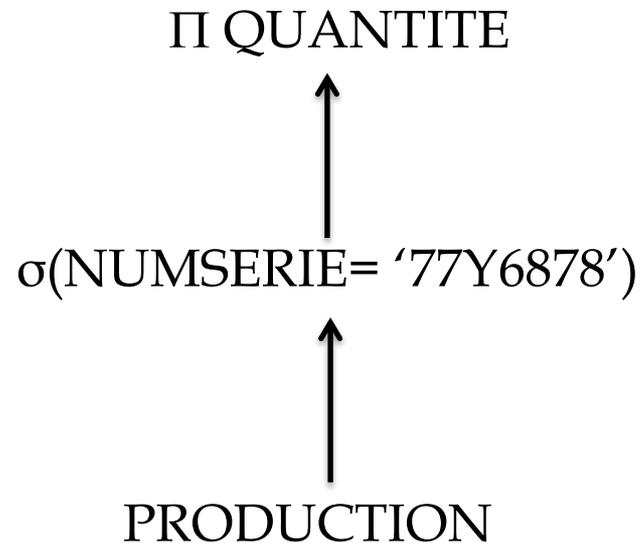
➤ Requête SQL:

```
SELECT QUANTITE  
FROM PRODUCTION  
WHERE NUMSERIE= '77Y6878';
```

Exercice 02

R1 : Quelle est la quantité disponible du produit de numéro de série '77y6878'.

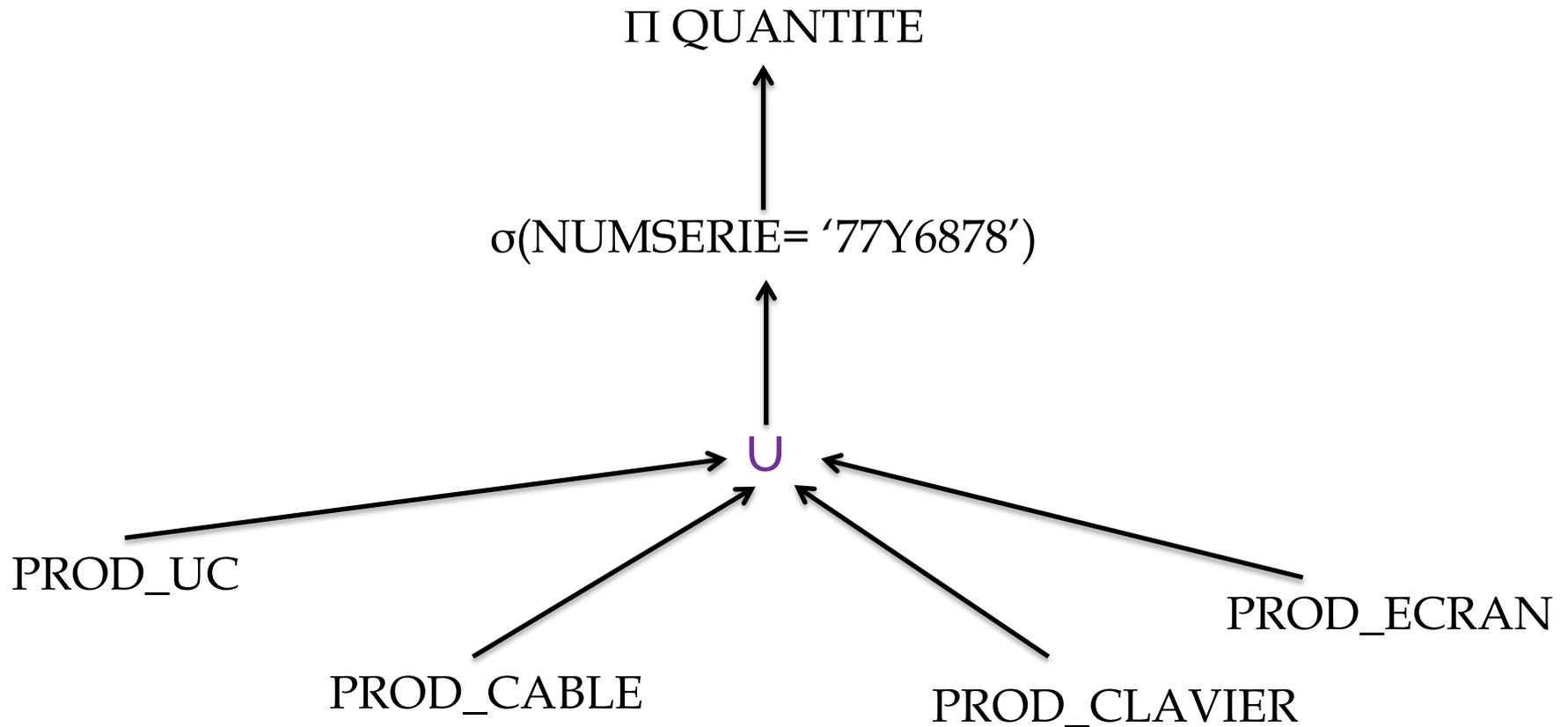
➤ Arbre global:



Exercice 02

R1 : Quelle est la quantité disponible du produit de numéro de série '77y6878'.

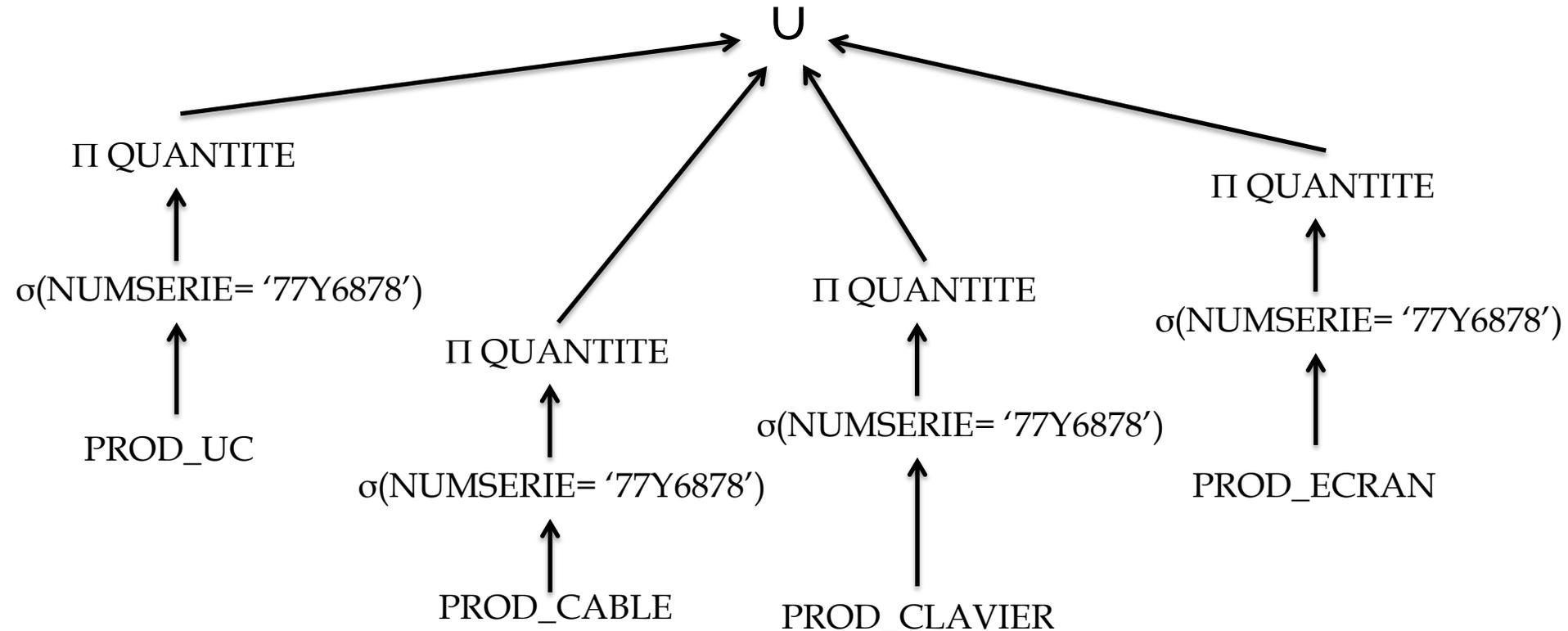
➤ Arbre canonique:



Exercice 02

R1 : Quelle est la quantité disponible du produit de numéro de série '77y6878'.

➤ Règles d'optimisation générales



Exercice 02

R2 : La somme des quantités de produits par composant et modèle.

➤ Requête SQL:

```
SELECT SUM(QUANTITE), COMPOSANT, MODELE  
FROM PRODUCTION  
GROUP BY COMPOSANT, MODELE
```

Exercice 02

R2 : La somme des quantités de produits par composant et modèle.

➤ Arbre global:

Π SUM(QUANTITE)
COMPOSANT, MODELE

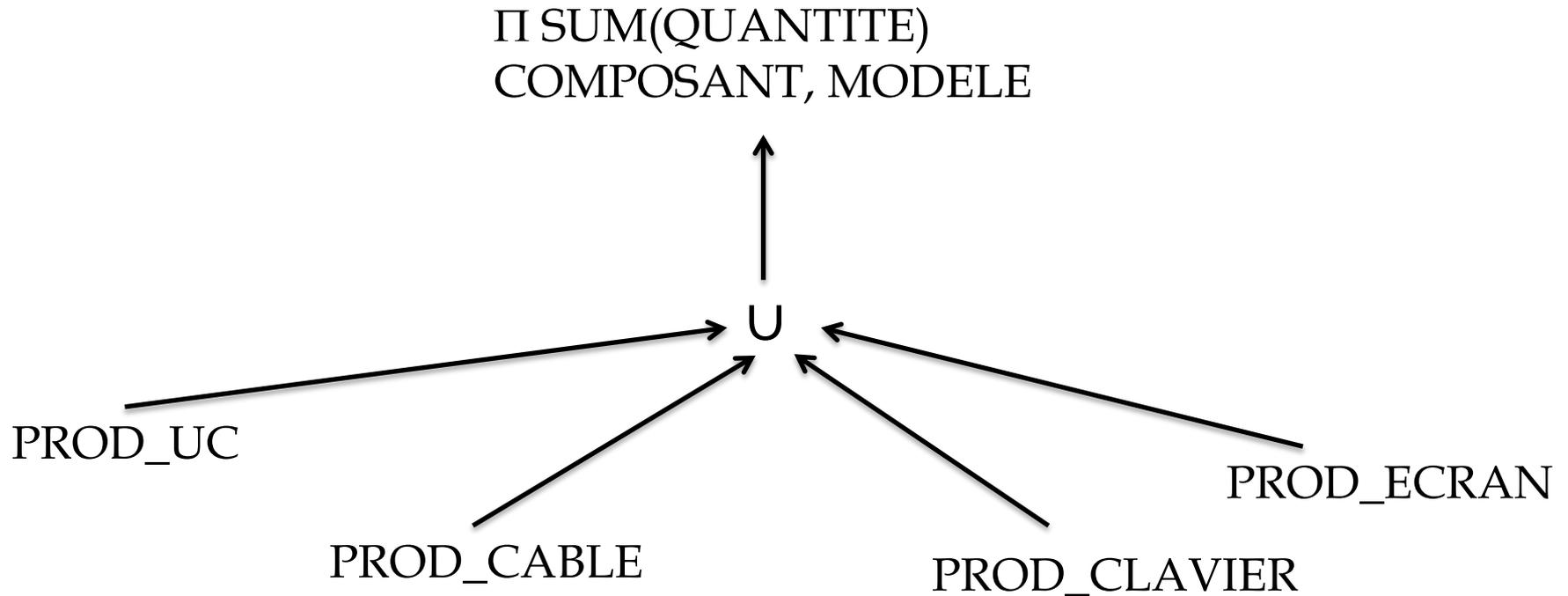


PRODUCTION

Exercice 02

R2 : La somme des quantités de produits par composant et modèle.

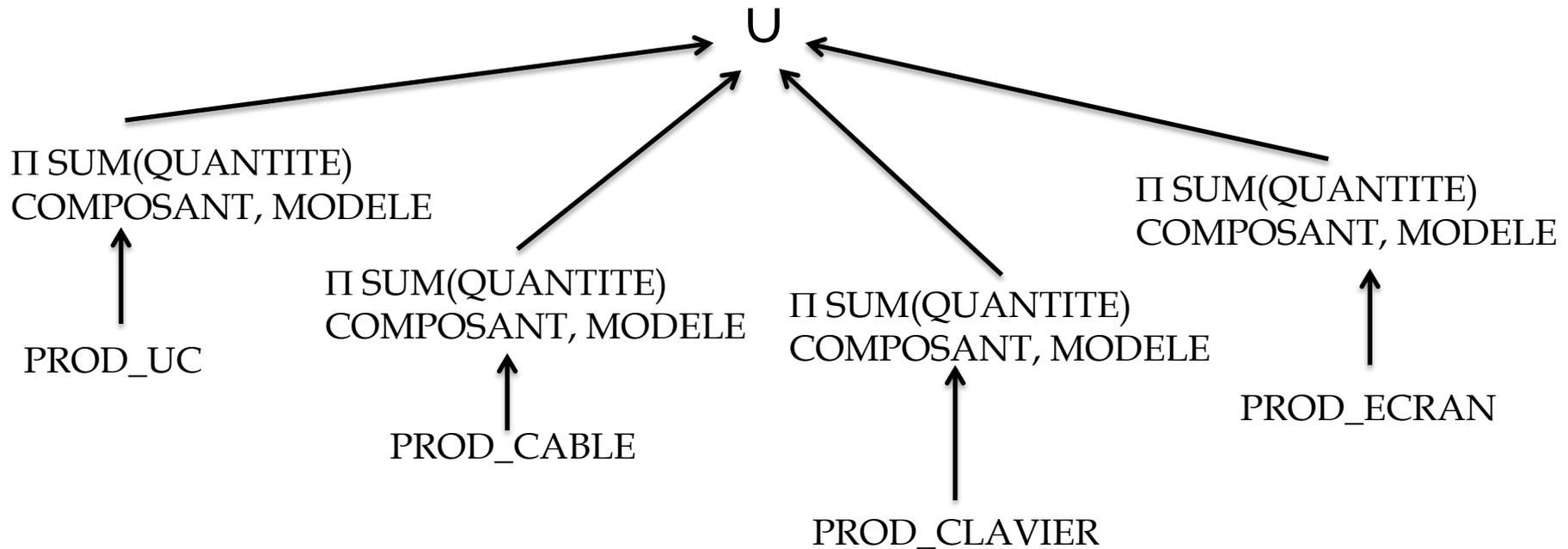
➤ Arbre canonique:



Exercice 02

R2 : La somme des quantités de produits par composant et modèle.

➤ Règles d'optimisation générales



Exercice 02

R3 : Quelle est la quantité disponible des unités centrales.

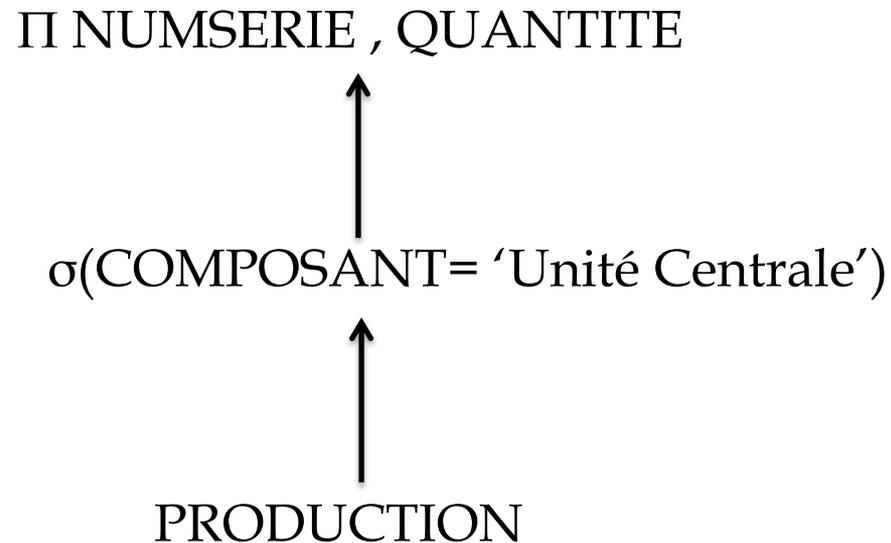
➤ Requête SQL:

```
SELECT NUMSERIE , QUANTITE  
FROM PRODUCTION  
WHERE COMPOSANT= 'Unité Centrale';
```

Exercice 02

R3 : Quelle est la quantité disponible des unités centrales.

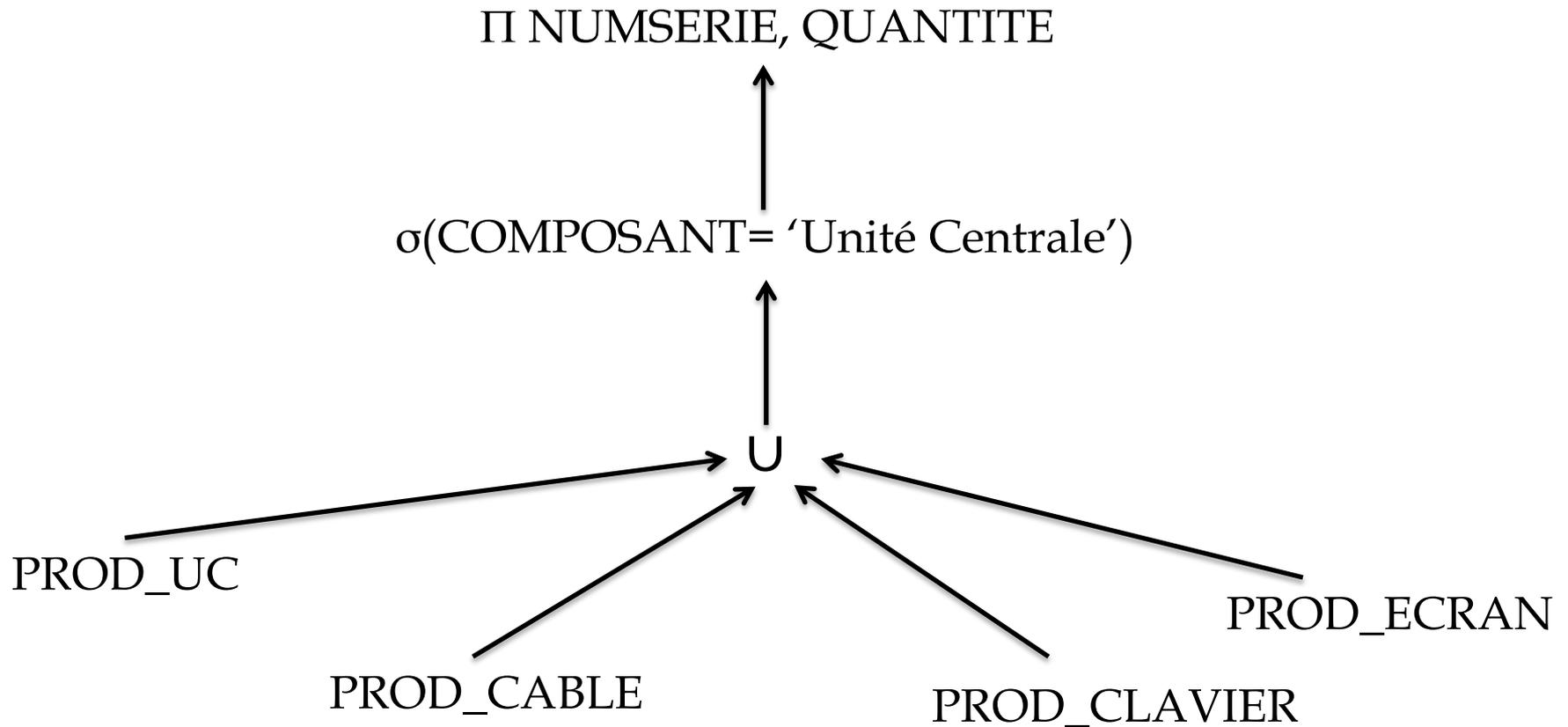
➤ Arbre global:



Exercice 02

R3 : Quelle est la quantité disponible des unités centrales.

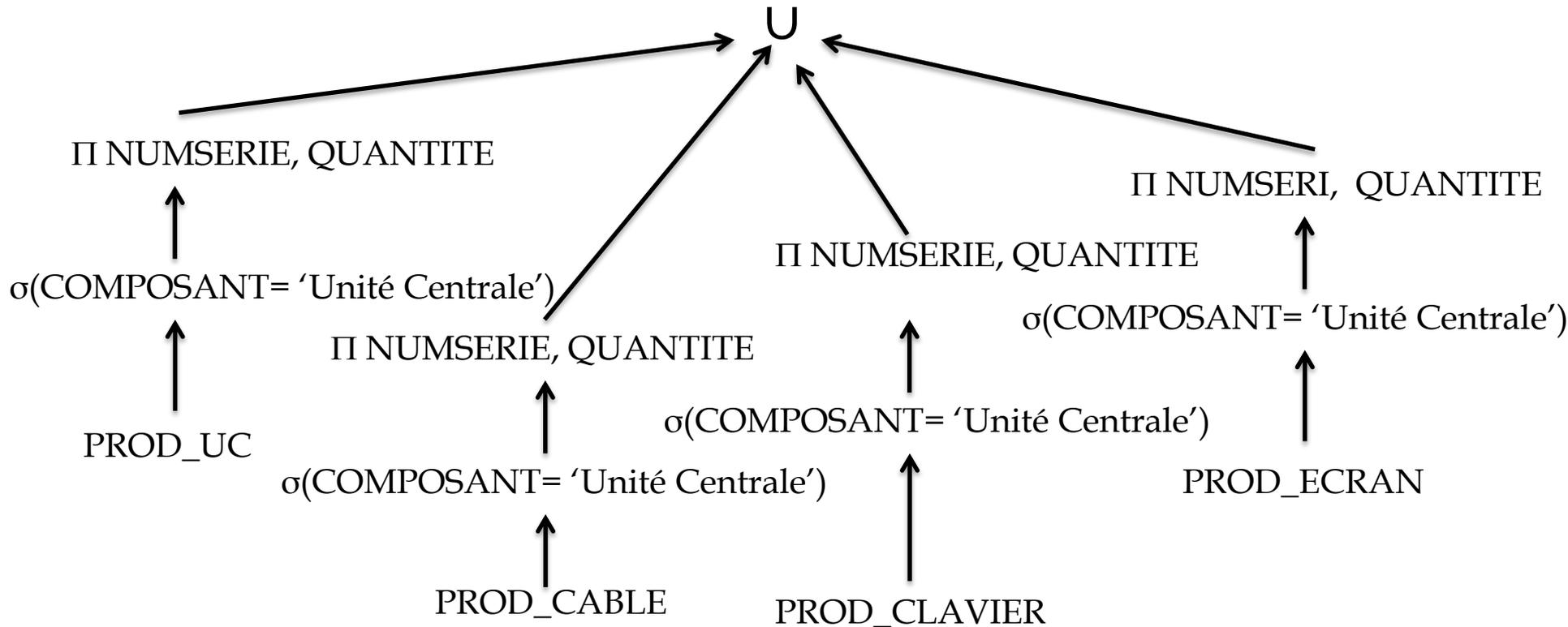
➤ Arbre canonique:



Exercice 02

R3 : Quelle est la quantité disponible des unités centrales.

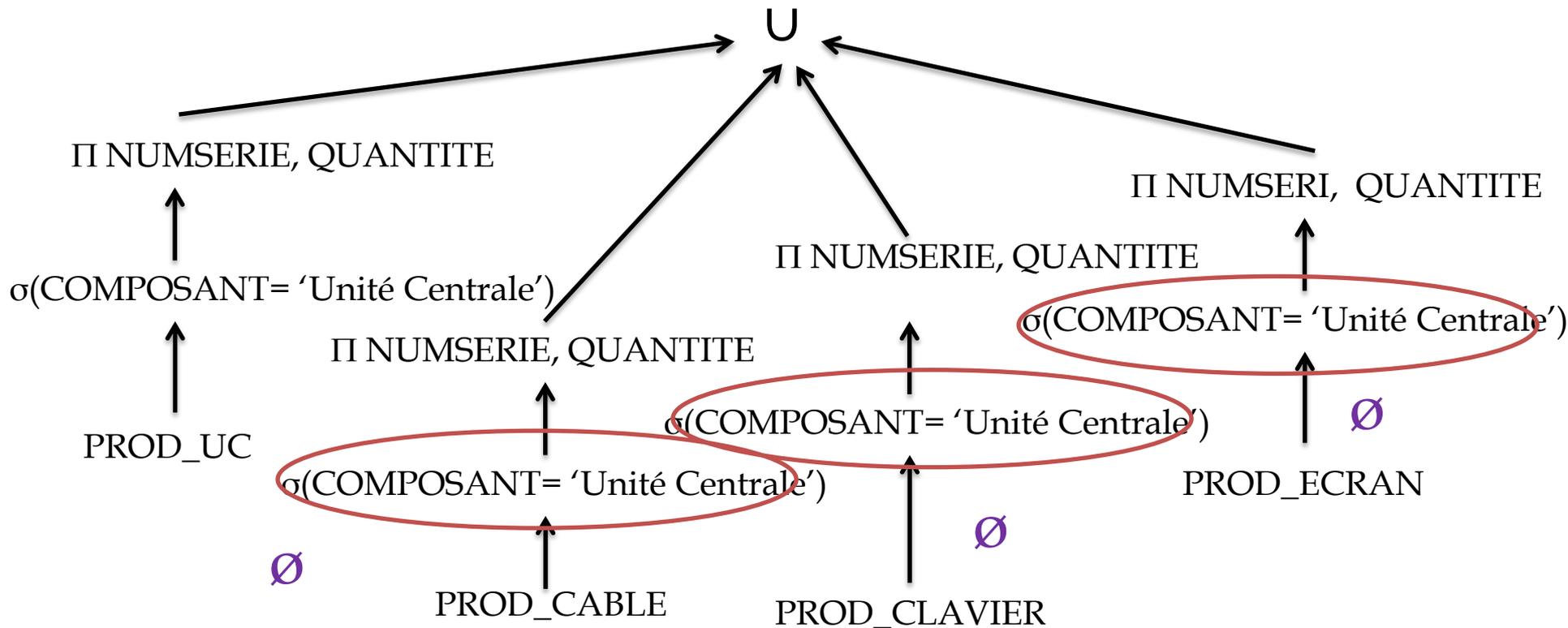
➤ Règles d'optimisation générales



Exercice 02

R3 : Quelle est la quantité disponible des unités centrales.

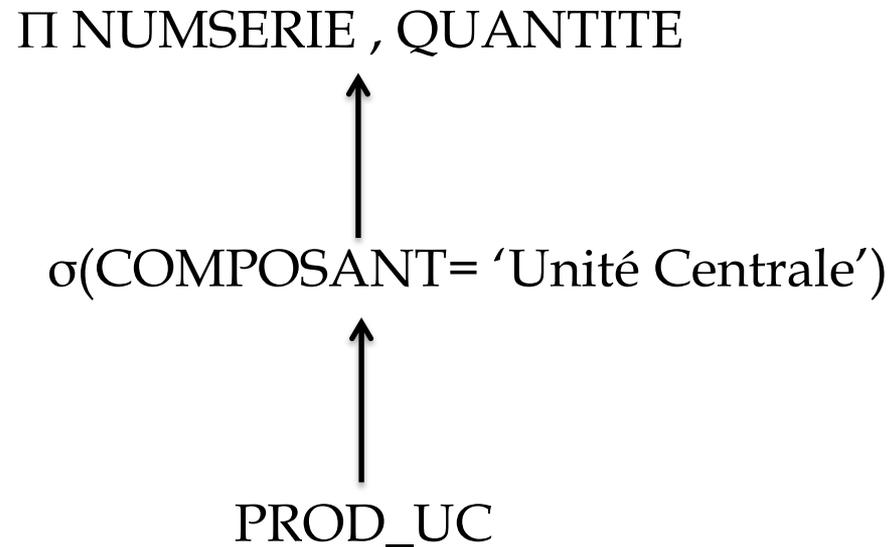
➤ Eliminer les fragments vides



Exercice 02

R3 : Quelle est la quantité disponible des unités centrales.

➤ Arbre optimisé:



Exercice 02

R3 : Quelle est la quantité disponible des unités centrales.

➤ Arbre optimisé:

Π NUMSERIE , QUANTITE



PROD_UC

Exercice 02

- b. Donner les différentes stratégies d'exécution de la requête R2 qui est lancée à Annaba

Stratégie	Description
Stratégie 01	Envoyer les fragments PROD_UC, PROD_CLAVIER et PROD_CABLE à Annaba et exécuter la requête
Stratégie 01	Envoyer les projections sur les attributs modèle et quantité pour chaque fragments PROD_UC, PROD_CLAVIER et PROD_CABLE à Annaba et exécuter la requête.
Stratégie 03	Calculer la somme des quantités de produits par composant et modèle pour chaque site et envoyer le résultat à Annaba et exécuter l'union avec le résultat des composants de Annaba.
Stratégie 04	Calculer la somme des quantités de produits par composant et modèle pour les sites Tarf et Sétif puis envoyer le résultat à Alger et exécuter l'union avec la somme des quantités de produits par composant et modèle pour site Alger. Envoyer le résultat à Annaba et faire l'union avec le résultat des composants de Annaba.

Exercice 03

❑ Une école de conduite établie à Alger depuis plusieurs années n'a pas cessé de se développer sur le territoire national au point que ses dirigeants décident de créer des agences au niveau des villes de Constantine et d'Oran.

❑ La base de données centralisée qui avait été développée initialement a la structure suivante :

- AGENCE (NumAG, VilleAG, Num-gerant, Nbr-instructeurs)
- PERSONNEL-ADMINISTRATIF (NumP, AdresseP, NomP, NumAG, Date-naissance, Fonction)
- VEHICULE (NumV, Marque, Type, NumAG, Num-Instructeur)
- CLIENT (NumCli, NomCli, AdresseCli, NumAG, Age)
- COURS (Num-Instructeur, NumCli, Date-cours, NumV)
- EXAMEN (NumCli, Date-examen, Resultat)

❑ Remarques

Le gérant et les instructeurs font partie du personnel administratif

Un véhicule est affecté à un instructeur

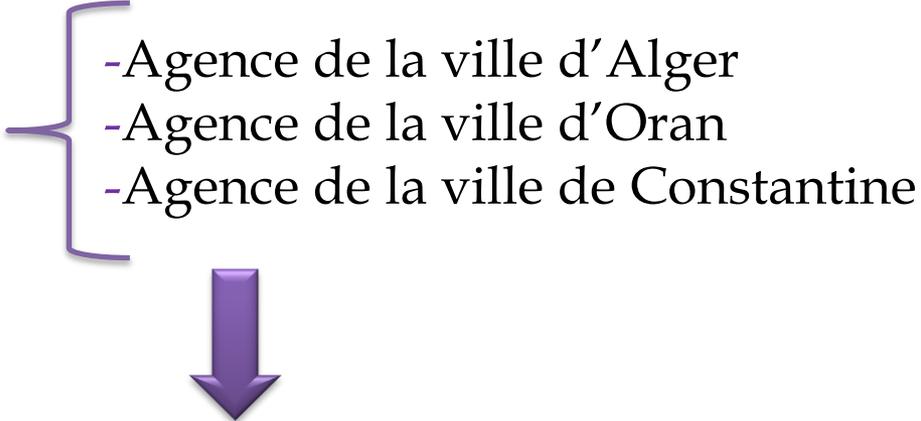
L'école veut créer trois sites pour héberger les données relatives aux agences de chaque ville : Site d'Alger, Site d'Oran et Site de Constantine.

Exercice 03

1. Fragmentation

- a. Proposer un schéma de fragmentation de cette base de données en tenant compte des hypothèses sur les sites. Donnez les expressions algébriques pour chaque table.

➤ AGENCE (NumAG, VilleAG, Num-gerant, Nbr-instructeurs)

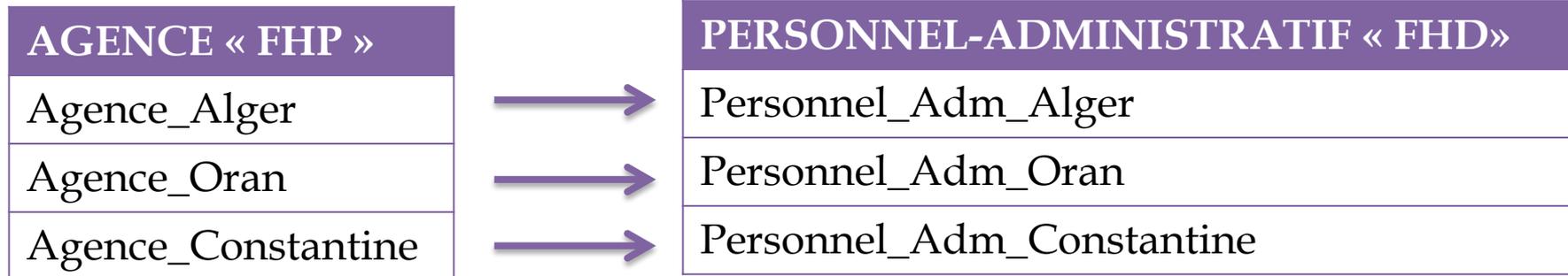
- 
- Agence de la ville d'Alger
 - Agence de la ville d'Oran
 - Agence de la ville de Constantine

Fragmentation horizontale primaire « **FHP** » de la table Agence sur l'attribut **Ville**

Exercice 03

1. Fragmentation

- a. Proposer un schéma de fragmentation de cette base de données en tenant compte des hypothèses sur les sites. Donnez les expressions algébriques pour chaque table.
 - PERSONNEL-ADMINISTRATIF (NumP, AdresseP, NomP, NumAG, Date-naissance, Fonction)

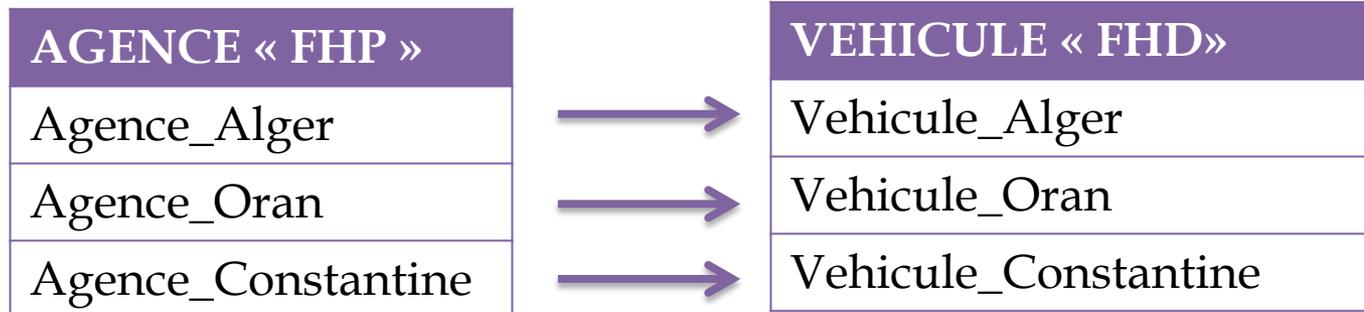


La table PERSONNEL-ADMINISTRATIF est fragmentée selon une fragmentation horizontale dérivée « **FHD** » de la table AGENCE donc on regroupera le personnel par rapport aux agences.

Exercice 03

1. Fragmentation

- a. Proposer un schéma de fragmentation de cette base de données en tenant compte des hypothèses sur les sites. Donnez les expressions algébriques pour chaque table.
- VEHICULE (NumV, Marque, Type, NumAG, Num-Instructeur)

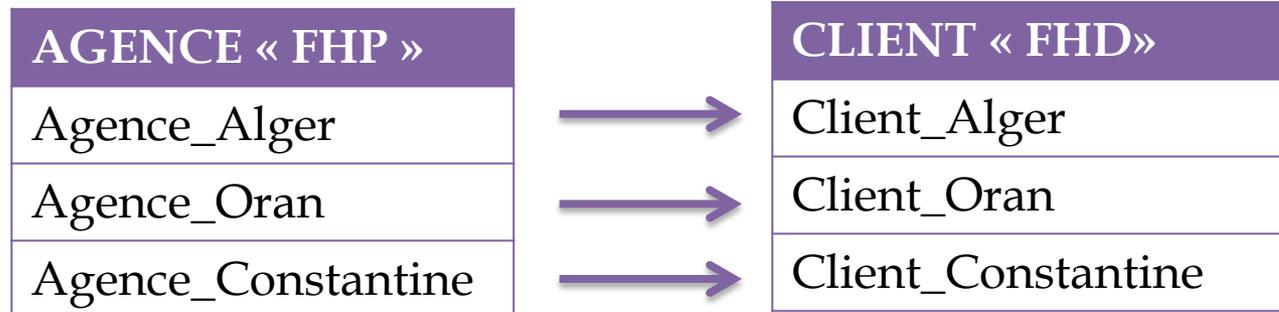


La table VEHICULE est fragmentée selon une fragmentation horizontale dérivée « **FHD** » de la table AGENCE donc on regroupera les véhicules par rapport aux agences.

Exercice 03

1. Fragmentation

- a. Proposer un schéma de fragmentation de cette base de données en tenant compte des hypothèses sur les sites. Donnez les expressions algébriques pour chaque table.
- CLIENT (NumCli, NomCli, AdresseCli, NumAG, Age)

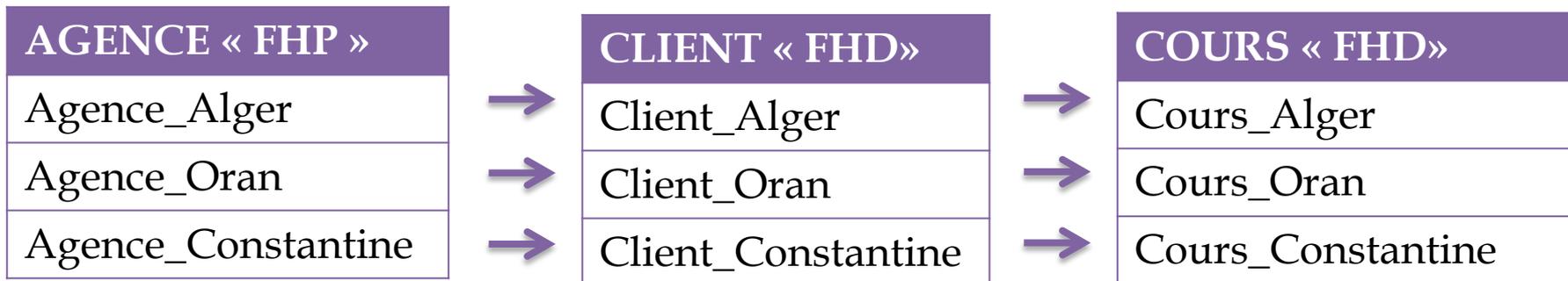


La table CLIENT est fragmentée selon une fragmentation horizontale dérivée « **FHD** » de la table AGENCE donc on regroupera les clients par rapport aux agences.

Exercice 03

1. Fragmentation

- a. Proposer un schéma de fragmentation de cette base de données en tenant compte des hypothèses sur les sites. Donnez les expressions algébriques pour chaque table.
 - COURS (Num-Instructeur, NumCli, Date-cours, NumV)



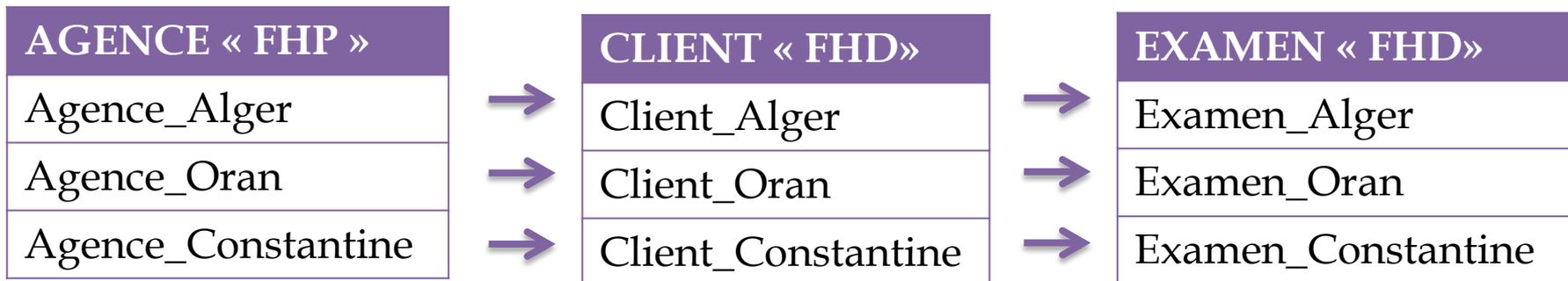
La table COURS est fragmentée selon une fragmentation horizontale dérivée « **FHD** » de la table CLIENT donc on regroupera les cours par rapport aux clients.

Exercice 03

1. Fragmentation

- Proposer un schéma de fragmentation de cette base de données en tenant compte des hypothèses sur les sites. Donnez les expressions algébriques pour chaque table.

➤ EXAMEN (NumCl, Date-examen, Resultat)



La table EXAMEN est fragmentée selon une fragmentation horizontale dérivée « **FHD** » de la table CLIENT donc on regroupera les examens par rapport aux clients.

Exercice 03

a. Expressions algébriques des fragment

Table	Fragment/Expression Algébrique
Agence « FHP »	Agence_Alger = $\sigma_{\text{ville=Alger}}$ (Agence) Agence_Oran = $\sigma_{\text{ville=Oran}}$ (Agence) Agence_Constantine = $\sigma_{\text{ville=Constantine}}$ (Agence)
Personnel-Administratif « FHD »	Personnel_Adm_Alger = Personnel Administratif \bowtie Agence_Alger Personnel_Adm_Oran = Personnel Administratif \bowtie Agence_Oran Personnel_Adm_Constantine = Personnel Administratif \bowtie Agence_Constantine
Véhicule « FHD »	Vehicule_Alger = Vehicule \bowtie Agence_Alger Vehicule_Oran = Vehicule \bowtie Agence_Oran Vehicule_Constantine = Vehicule \bowtie Agence_Constantine
Client « FHD »	Client_Alger = Client \bowtie Agence_Alger Client_Oran = Client \bowtie Agence_Oran Client_Constantine = Client \bowtie Agence_Constantine
Examen « FHD »	Examen_Alger = Examen \bowtie Client_Alger Examen_Oran = Examen \bowtie Client_Oran Examen_Constantine = Examen \bowtie Client_Constantine
COURS « FHD »	Cours_Alger = Cours \bowtie Client_Alger Cours_Oran = Cours \bowtie Client_Oran Cours_Constantine = Cours \bowtie Client_Constantine

Exercice 03

1. Fragmentation

- b. Quelles sont les possibilités de fragmentation de la table COURS. Donner la meilleure solution.

➤ COURS (Num-Instructeur, NumCli, Date-cours, NumV)

Solution 1	FHD de la table PERSONNEL-ADMINISTRATIF
Solution 2	FHD de la table CLIENT
Solution 3	FHD de la table VEHICULE
Meilleure solution	Les trois solutions mènent au même résultat car les tables VEHICULE, CLIENT ET PERSONNEL-ADMINISTRATIF sont fragmentée sur la même clé étrangère NumAG.

Exercice 03

1. Fragmentation

- c. Donner les requêtes SQL correspondant à la fragmentation des deux relations AGENCE et CLIENT.

➤ Table Agence

```
Create table AGENCE (NumAG integer , VilleAG varchar(50), Num-  
gerant integer, Nbr-instructeurs integer, constraint pk_agence  
primary key (NumAG ))
```

Partition by List (VilleAG)

```
(  
Partition AGENCE_Alger VALUES ('Alger'),  
Partition AGENCE_Oran VALUES ('Oran'),  
Partition AGENCE_Constantine VALUES ('Constantine')  
);
```

Exercice 03

1. Fragmentation

- c. Donner les requêtes SQL correspondant à la fragmentation des deux relations AGENCE et CLIENT.

➤ **Table Client**

```
Create table CLIENT (NumCli integer , NomCli varchar(50), AdresseCli
varchar(100), NumAG integer, Age integer,
constraint pk_client primary key (NumCli ),
constraint fk_agence foreign key(NumAG) refernces agence(NumAG))
Partition by reference (fk_Agence);
```

Exercice 03

1. Fragmentation

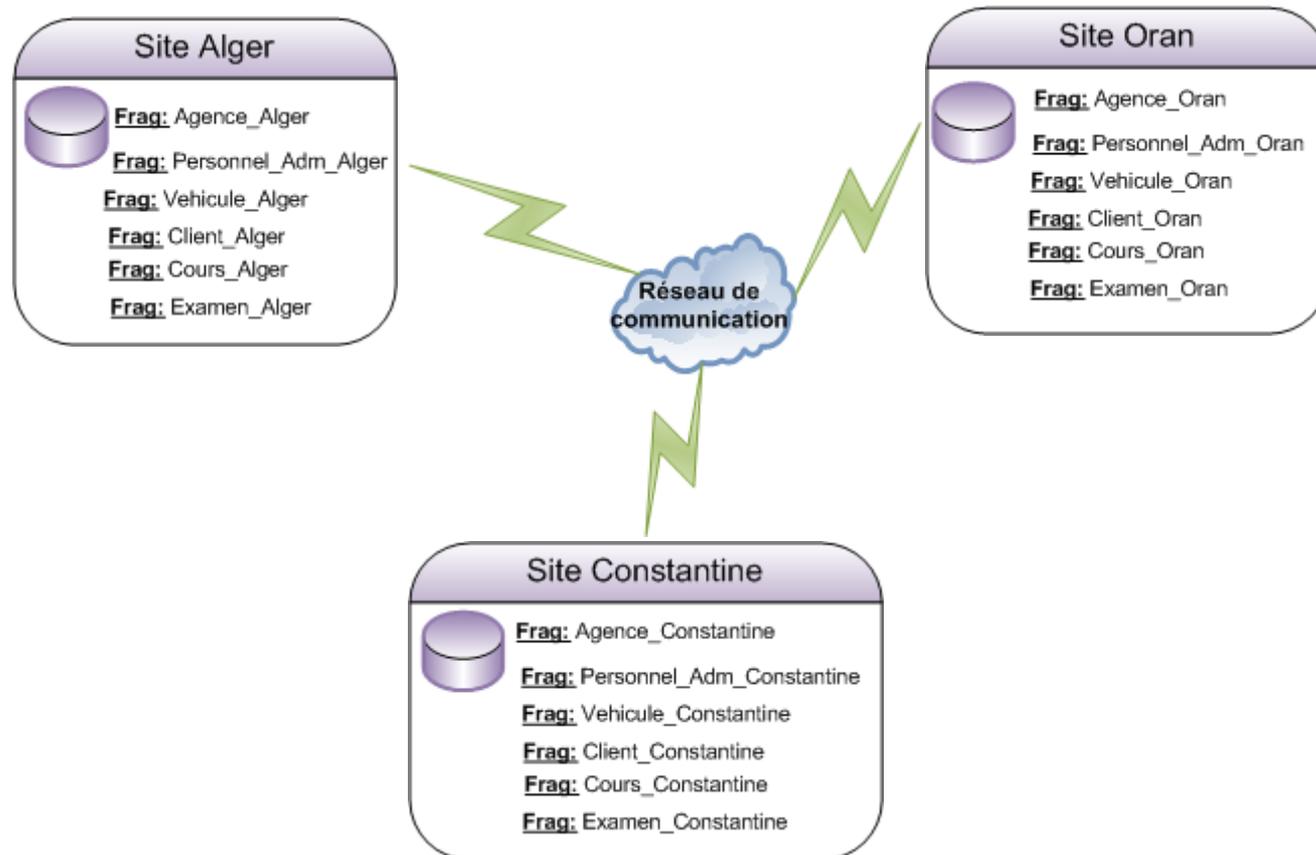
- d. Oracle fragmente une table dès sa création. Donner une solution pour fragmenter une table contenant déjà des données. Appliquer la solution sur la table AGENCE.

Solution	Application sur la table AGENCE
Créer une table Agence_Frag fragmentée ayant la même structure	Voir script réponse précédente
Transférer les données dans la table fragmentée Agence_Frag	Insert into Agence_Frag select * from Agence
Supprimer la table initiale	Drop Table Agence
Renommer la table fragmentée Agence_Frag	Alter Table Agence_Frag rename to Agence

Exercice 03

2. Allocation

- a. Proposer un schéma d'allocation de tous les fragments.



Exercice 03

2. Allocation

- b. Les véhicules appartenant à des agences de la wilaya d'Alger, sont très souvent accédés à partir d'Oran. Proposer une amélioration du schéma d'allocation en prenant en compte cette contrainte. Donner les inconvénients de cette solution.

Solution	Inconvénient
Répliquer (créer une copie) du fragment Vehicule_Alger au site d'Oran.	Problèmes de mise à jour. A chaque fois qu'un changement est fait sur Alger ou Oran, il faut mettre à jour l'autre fragment en temps réel.

Exercice 03

3. Evaluation des requêtes distribuées

Sachant que la fragmentation est transparente à l'utilisateur et que la requête **R** suivante est émise à partir du site d'Alger : « Donner les nom et numéro de tous les clients de moins de 25 ans, relevant des agences d'Alger ou d'Oran, qui ont eu un résultat positif à l'examen du 22/05/2019»

- a. Donner le script SQL associé, l'arbre algébrique global, l'arbre canonique ainsi que l'arbre algébrique simplifié.
- b. Donner deux stratégies d'exécution de la requête **R**

Exercice 03

3. Evaluation des requêtes distribuées

a. Donner le script SQL associé, l'arbre algébrique global, l'arbre canonique ainsi que l'arbre algébrique simplifié.

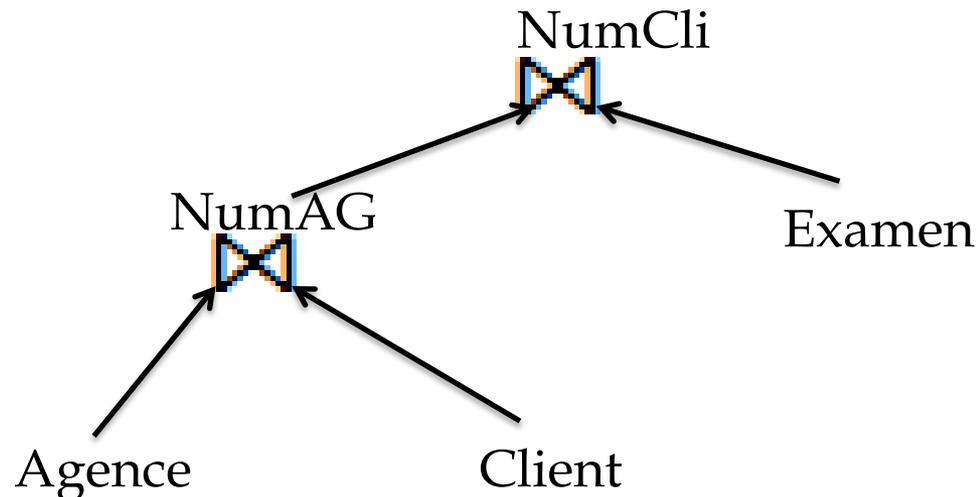
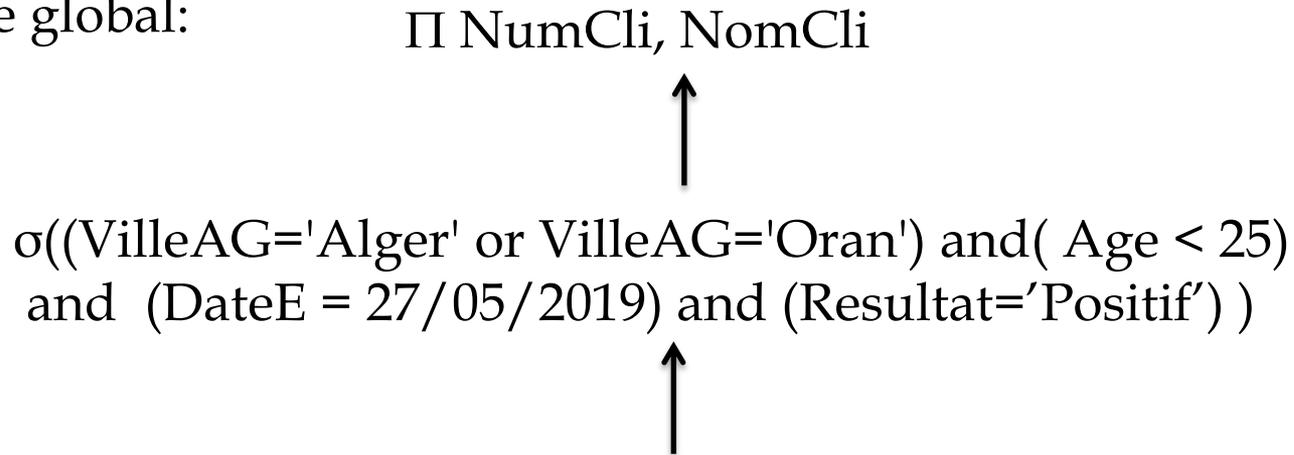
➤ Requête SQL:

```
Select NumCli, NomCli  
From AGENCE A, CLIENT C, EXAMEN E  
Where (A.VilleAG='Alger' or A.VilleAG='Oran')  
and (C.Age < 25)  
and (E.DateE = 27/05/2019)  
and (E.Resultat='Positif')  
and (C.NumAG = A.NumAG )  
and (C.NumCli = E.NumCli );
```

Exercice 03

3. Evaluation des requêtes distribuées

➤ Arbre global:

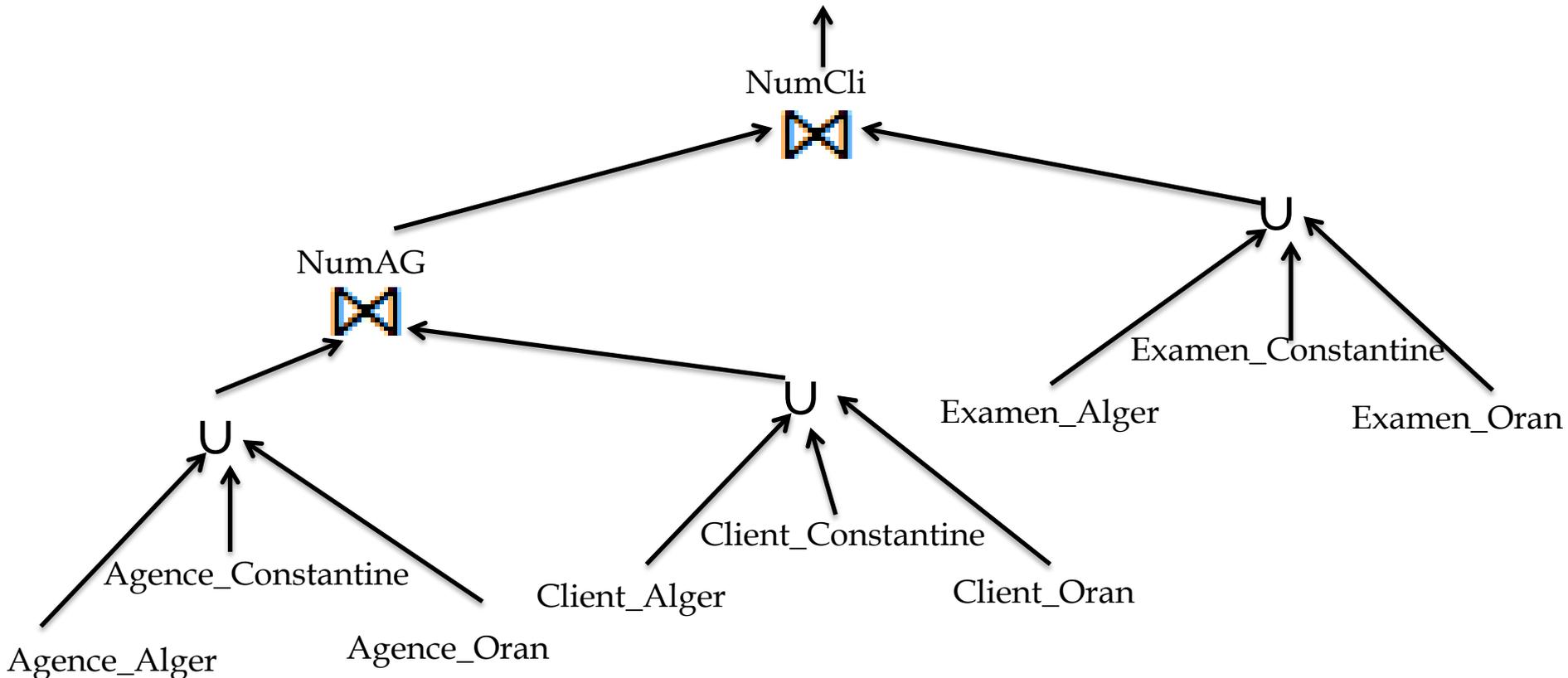


Exercice 03

➤ Arbre canonique:

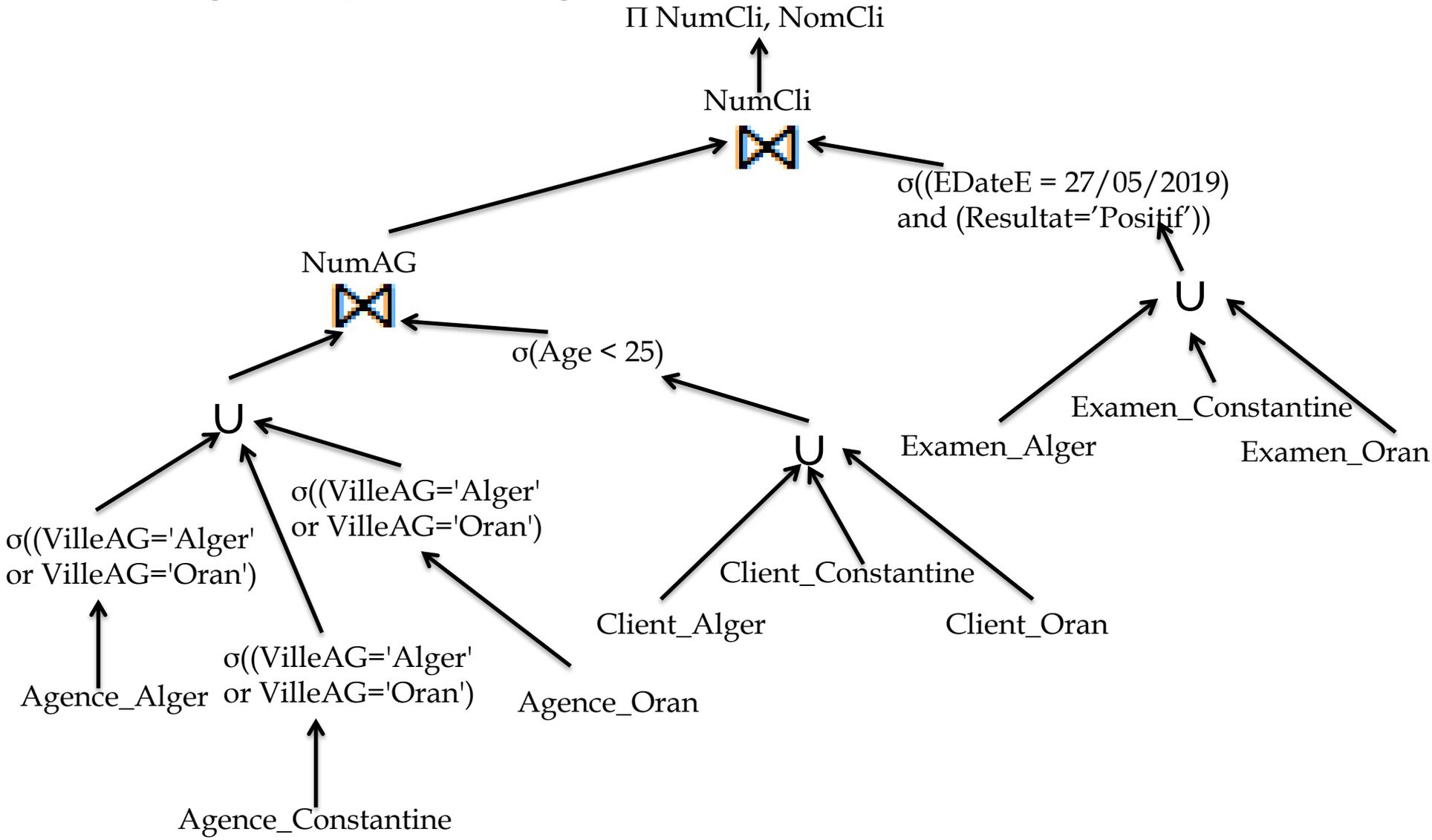
$\Pi \text{ NumCli, NomCli}$

$\sigma((\text{VilleAG}='Alger' \text{ or } \text{VilleAG}='Oran') \text{ and } (\text{Age} < 25) \text{ and } (\text{DateE} = 27/05/2019) \text{ and } (\text{Resultat}='Positif'))$



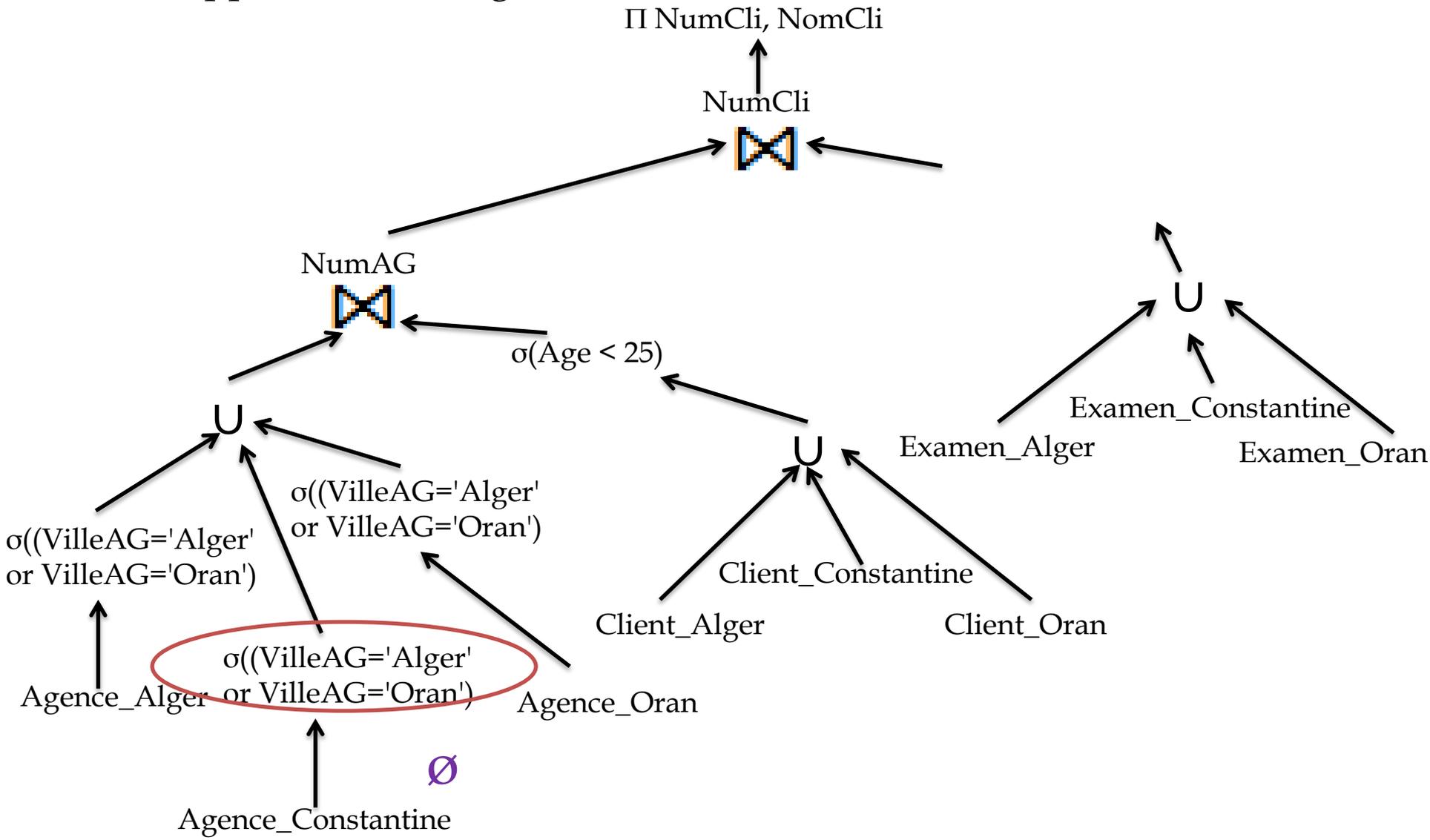
Exercice 03

- Règles d'optimisation générales:



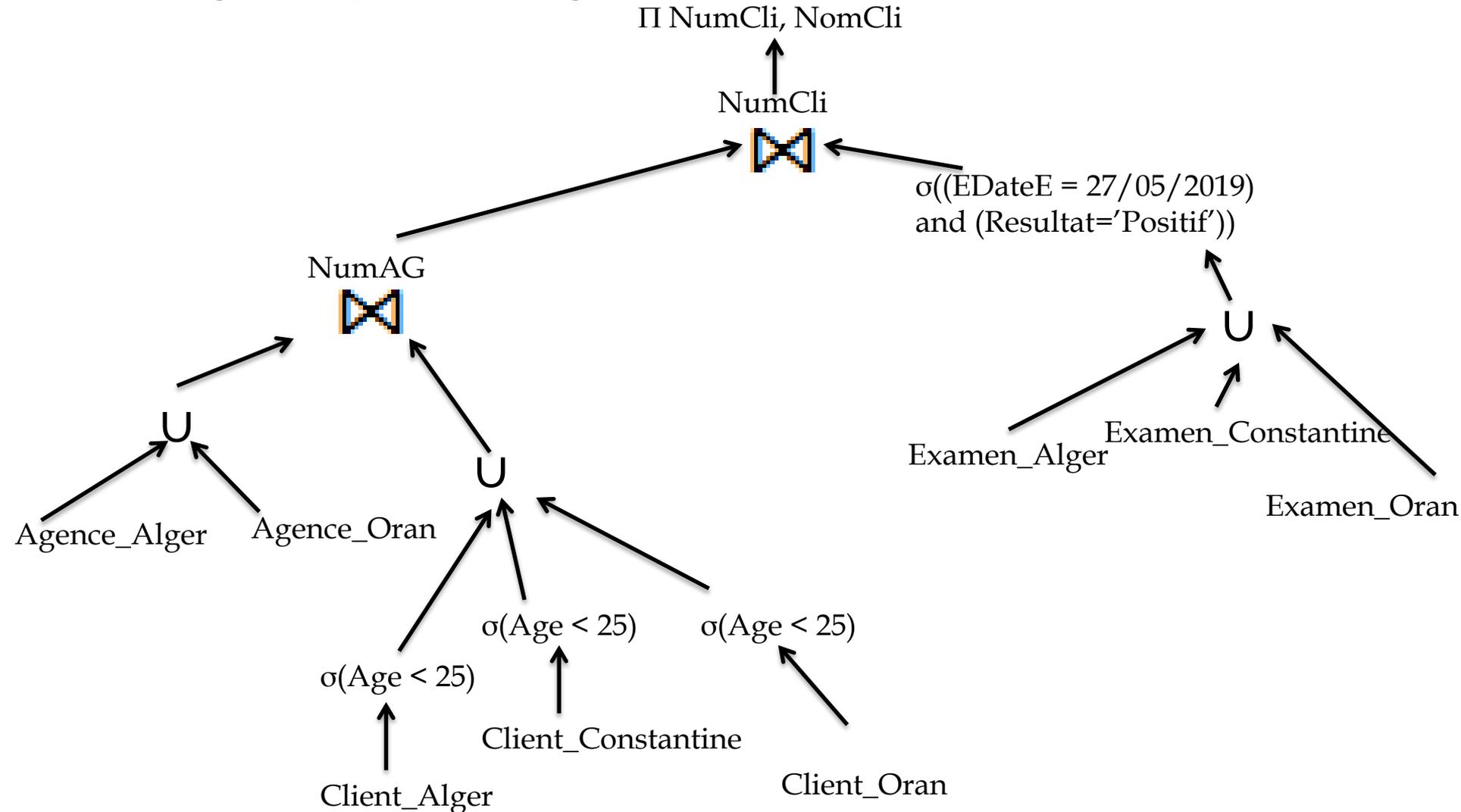
Exercice 03

- Suppression de fragment vide:



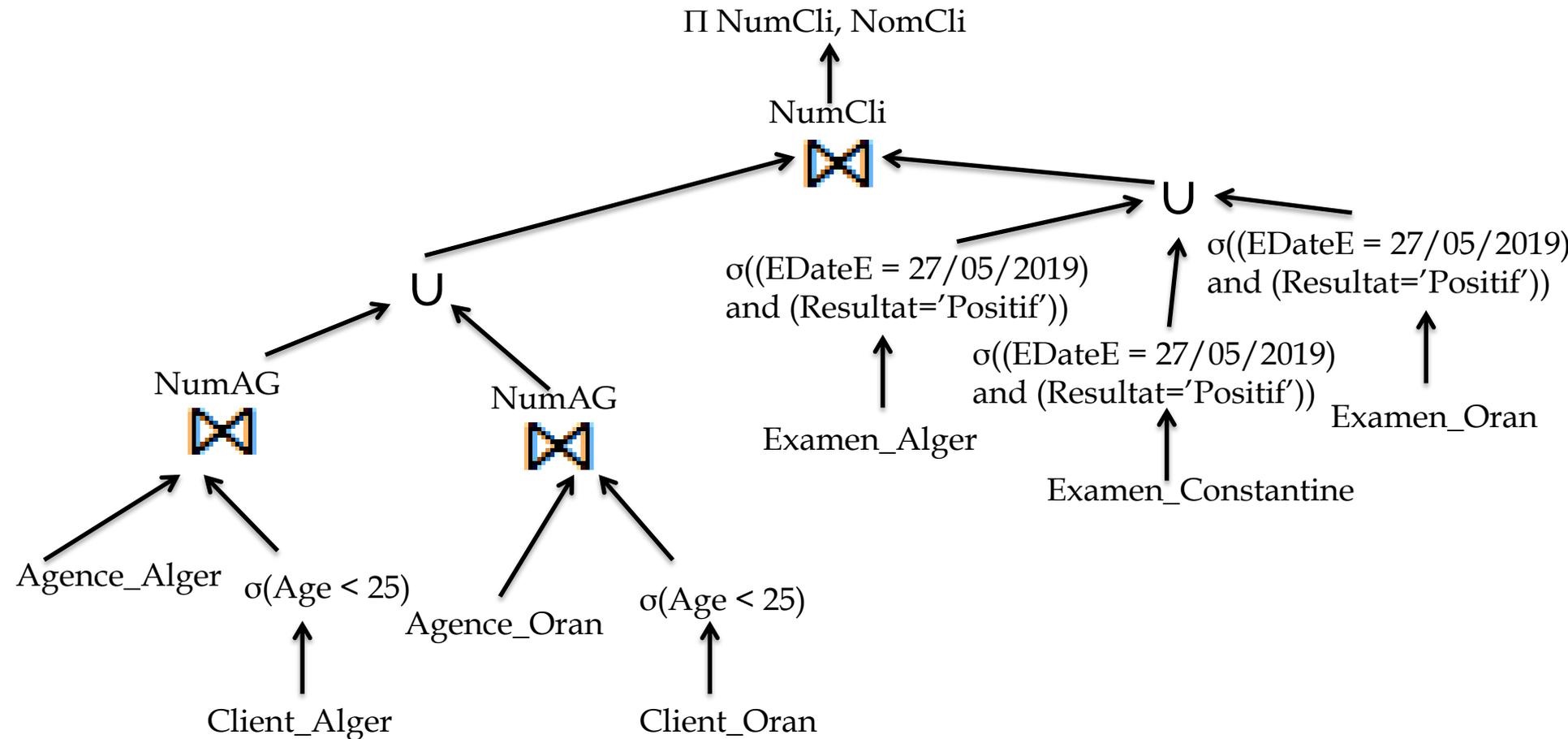
Exercice 03

- Règles d'optimisation générales:



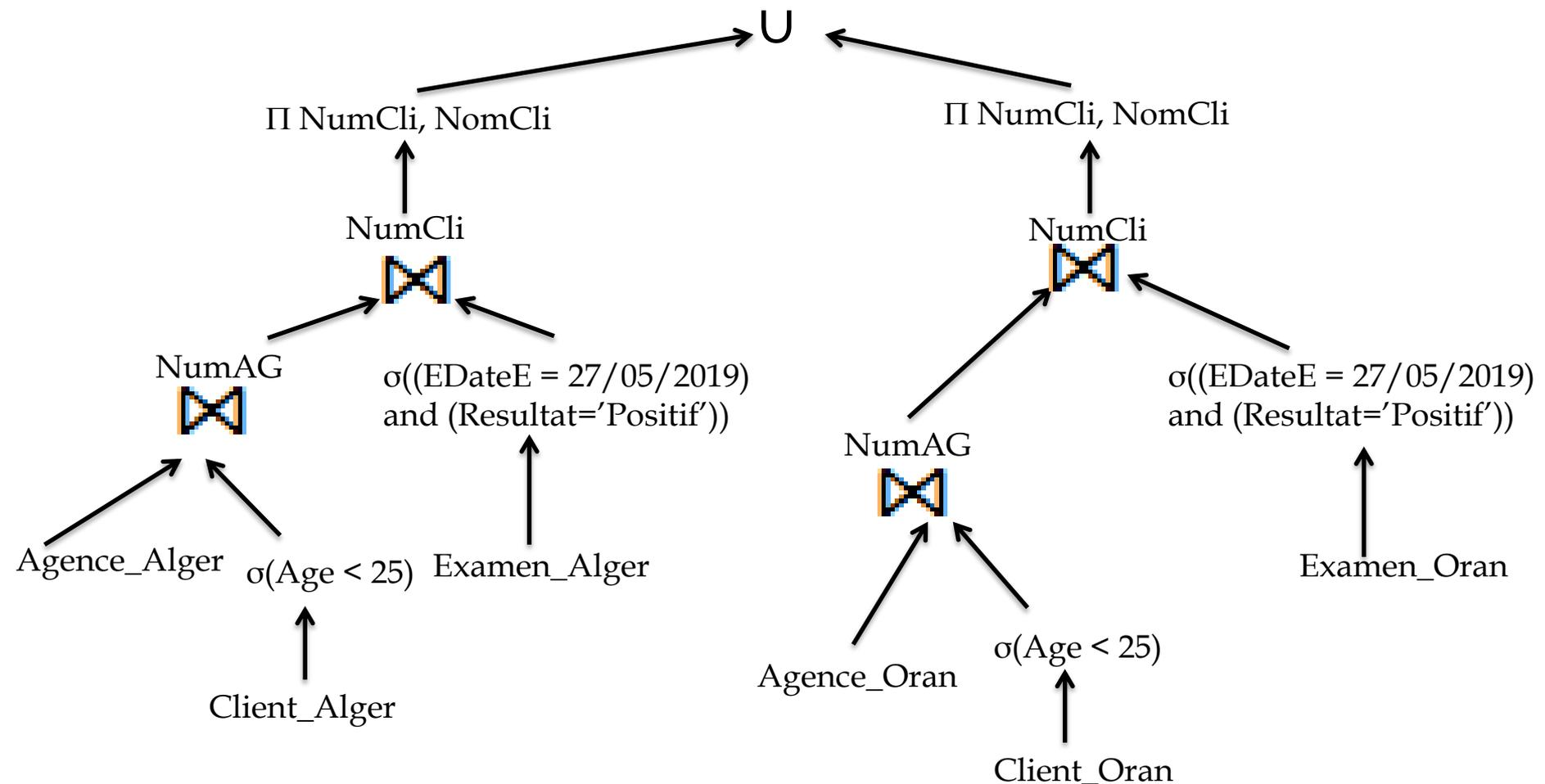
Exercice 03

- Distribution de la jointure par rapport à l'union et élimination des fragments vides et règles d'optimisation générales :



Exercice 03

- Distribution de la jointure par rapport à l'union et élimination des fragments vides:



Exercice 03

- b. Donner deux stratégies d'exécution de la requête R

Stratégie	Description
Stratégie1	Exécuter la partie gauche de l'arbre précédent à Alger Exécuter la partie droite à Oran et envoyer le résultat à Alger Faire l'union des deux résultats
Stratégie 2	Envoyer les fragments Agence_Oran, Client_Oran et Examen_Oran à Alger Exécuter toute la requête sur Alger