

## Solution TD3 : Algèbre relationnelle

Soit la base de données relationnelle PUF, de schéma :

U (NU, NomU, Ville)

P (NP, NomP, Couleur, Poids)

F (NF, NomF, Statut, Ville)

PUF (NP, NU, NF, Quantité)

NP référence P.NP

NU référence U.NU

NF référence F.NF

**Exprimer en algèbre relationnelle les requêtes suivantes :**

- 1) Donner le numéro, le nom et la ville de toutes les usines :

$\Pi$  [NU, NomU, Ville]U

// la projection son équivalent en SQL :

SELECT NU, NomU, Ville //les attributs désirés dans le résultat

FROM U ; // la table sur laquelle se fait la projection

- 2) Donner le numéro, le nom et la ville de toutes les usines d'Annaba:

$\Pi$  [NU, NomU, Ville]U  $\sigma$  [Ville="annaba"]U

//la sélection en SQL : on utilise la clause WHERE suivie de la condition WHERE Ville = "Annaba"

DONC la requête entière sera:

SELECT NU, nomU, Ville // les attributs désirés dans le résultat lorsque on veut tous les attributs de la  
// table on peut mettre \*

FROM U // la table sur laquelle on travaille

WHERE Ville = "Annaba"; // la condition de la sélection (restriction)

- 3) Donner les numéros des fournisseurs qui approvisionnent les usines numéro1 en produit numéro1 :

$\Pi$  [NF]  $\sigma$  [NU=1 et NP=1]PUF

- 4) Donner le nom et la couleur des produits livrés par le fournisseur n°1 :

$\Pi$ [NomP, Couleur] P  $\bowtie$   $\sigma$  [NF=1] PUF

- 5) Donner les numéros des fournisseurs qui approvisionnent l'usine n°1 en un produit rouge :

$\Pi$ [NF]  $\sigma$  [NU=1 et Couleur= "rouge"] P  $\bowtie$  PUF (jointure naturelle) ou bien

$\Pi$ [NF] ( $\sigma$  [Couleur= "rouge"] P)  $\bowtie$  ( $\sigma$  [NU=1] PUF) (//, req optimisée)

// Jointure en SQL les deux tables seront après la clause FROM séparées par une "," (/ JOIN à ON)

// la condition de jointure doit être écrite explicitement ici: P.NP=PUF.NP (NomTable.NomAttribut)

SELECT NF

FROM P, PUF // Jointure (produit cartésien: s'il n'y a pas condition de jointure)

WHERE P.NP=PUF.NP AND //condition de jointure

NU=1 AND Couleur= "rouge" // condition de la requête (de la restriction)

Ou bien avec la thêta jointure (il faut renommer)

$$P1 = \alpha [NP : NP1] P$$

$$\Pi[NF] (\sigma [NU=1] PUF) \underset{[NP=NP1]}{\bowtie} (\Pi[NP1] \sigma [Couleur = "rouge"] P1)$$

6) Donner les noms des fournisseurs qui approvisionnent une usine d'Annaba ou d'Alger en un produit rouge :

$$\Pi[NomF] \sigma [(Ville = \text{Annaba} \text{ ou } Ville = \text{Alger}) \text{ et } Couleur = "rouge"] U \bowtie P \bowtie F \bowtie PUF \quad \text{ou bien}$$

$$\Pi[NomF] (\sigma [Ville = \text{Annaba} \text{ ou } Ville = \text{Alger}] U) \bowtie (\sigma [Couleur = "rouge"] P) \bowtie (\Pi[NF, NomF] F) \bowtie PUF$$

7) Donner les numéros des produits livrés à une usine par un fournisseur de la même ville :

$$\Pi[NP] PUF \bowtie (\Pi[NF, Ville] F) \bowtie (\Pi[NU, Ville] U)$$

8) Donner les numéros des produits livrés à une usine d'Alger par un fournisseur d'Alger:

$$\Pi[NP] PUF \bowtie (\sigma [Ville = "alger"] F) \bowtie U$$

9) Donner les numéros des usines qui ont au moins un fournisseur qui n'est pas de la même ville :

On doit renommer car il y a une condition d'inégalité entre les villes des fournisseurs et des usines

$$U1 = \alpha [NU : NU1, nomU : nomU1, Ville : Ville1] U$$

$$\Pi[NU1] (U1 \underset{[NU1=NU \text{ et } Ville1 \neq Ville]}{\bowtie} (PUF \underset{[NU=NU]}{\bowtie} F))$$

10) Donner les numéros des fournisseurs qui approvisionnent à la fois les usines n°1 et n°2 :

Solution 1 : Jointure sur une même table PUF  $\Rightarrow$  renommage obligatoire

$$PUF1 = \alpha (NP : NP1, NU : NU1, NF : NF1, Quantité : Quantité1) PUF$$

$$\Pi[NF] (\sigma [NU=1] PUF) \underset{[NF=NF1]}{\bowtie} \sigma [NU1=2] PUF1)$$

Solution 2: intersection

$$(\Pi[NF] \sigma [NU=1] PUF) \cap (\Pi[NF] \sigma [NU=2] PUF)$$

11) Donner les numéros des usines qui utilisent au moins un produit disponible chez le fournisseur n°3 (c'est-à-dire un produit qu'il livre mais pas nécessairement à cette usine) :

$$\Pi[NU] ( PUF \bowtie (\Pi [NP] \sigma [NF=3] PUF) )$$

12) Donner les **numéros des produits** qui sont livrés **à toutes** les usines d'Annaba :

Il s'agit d'une division : (**produit**, usine)/usine (restriction)=**produit** donc

$$\Pi[NP, NU] PUF / (\Pi[NU] (\sigma [Ville = \text{Annaba}] U))$$

13) Donner les numéros des usines qui s'approvisionnent uniquement chez le fournisseur n°3 :

Il s'agit de la soustraction

$$(\Pi[\text{NU}] \sigma [\text{NF}=3] \text{ PUF}) - (\Pi[\text{NU}] \sigma [\text{NF} \neq 3] \text{ PUF})$$

**Remarque 1 :** pour toutes les requêtes on peut répondre par étapes. Exemple avec la requête 13

$$R1 = \sigma [\text{NF}=3] \text{ PUF}$$

$$R2 = \Pi[\text{NU}] R1$$

$$R3 = \sigma [\text{NF} \neq 3] \text{ PUF}$$

$$R4 = \Pi[\text{NU}] R3$$

$$R5 = R2 \text{ } \ominus \text{ } R4$$

**Remarque 2 :** Ce TD sera réalisé en TP où vous allez créer une base de données -livraison- et traduire toutes les requêtes algébriques en SQL (TP3 LMD).

**Remarque 3 :** les opérateurs de division et de soustraction n'existent pas en SQL. L'opérateur d'intersection n'existe pas dans MySQL, il existe dans les autres SGBD comme Oracle et SQLServer.