Énoncé 3 : algèbre relationnelle

guillaume.postic@univ-evry.fr

Exercice 1

1. Oui
2. Oui
3. Non : union possible seulement si les deux relations sont de même type (même nombre d'attributs et mêmes domaines). En SQL, le nombre d'attributs est la seule condition.
4. Non : même raison que pour l'union (différence avec SQL)
5. Oui
6. Non (voir 3 et 4)
7. Oui
8. Oui : produit cartésien
9. Oui
10. Non : aucune colonne pour faire la jointure (différence avec SQL)
11. Oui : colonne NomF
12. Oui : comme 11 mais seul les attributs de Fournisseurs sont gardés
13. Oui
14. Oui
15. Oui
16. Oui
17. Oui
18. Oui
19. Oui

2. Formuler les requêtes suivantes en langage algébrique et en SQL :

Quels sont les noms des produits commandés par Jean ?

```
/*1*/
SELECT NomP FROM Commandes WHERE NomC = 'Jean'
```

$\pi_{\text{NomP}} (\sigma_{\text{NomC = 'Jean'}} (\textbf{Commandes}))$

Quels sont les noms des fournisseurs qui fournissent les produits qui figurent dans les commandes de Paul ?

```
/*2*/
SELECT p.NomF
FROM Commandes c
JOIN Prix p
ON c.NomP = p.NomP
WHERE c.NomC = 'Paul'
```

π_{NomF} (**Prix** $\bowtie$ $\sigma_{\text{NomC = 'Paul'}}$ (**Commandes**))

Quelle est l'adresse des fournisseurs qui fournissent des parpaings à un coût strictement inférieur à 1200 ?

```
/*3*/
SELECT f.AdresseF
FROM Prix p
JOIN Fournisseurs f
ON p.NomF = f.NomF
WHERE p.NomP = 'parpaing'
AND p.Couts < 1200
```

π_{AdresseF} (**Fournisseurs** $\bowtie$ $\sigma_{\text{NomP = 'parpaing'} \wedge \text{Couts} < 1200}$ (**Prix**))

Quels sont les noms et adresses des clients et des fournisseurs tels que le produit commandé lors d'une commande soit des briques ?

```
/*4*/
SELECT c.NomC, k.AdresseC, p.NomF, f.AdresseF
FROM Commandes c
JOIN Clients k
ON c.NomC = k.NomC
JOIN Prix p
ON c.NomP = p.NomP
JOIN Fournisseurs f
ON p.NomF = f.NomF
WHERE c.NomP = 'briques'
```

```
/* implicit join */
SELECT c.NomC, k.AdresseC, p.NomF, f.AdresseF
FROM Commandes c, Clients k, Prix p, Fournisseurs f
WHERE c.NomP = 'briques'
AND c.NomC = k.NomC
AND c.NomP = p.NomP
AND p.NomF = f.NomF
```

$\pi_{\text{AdresseF, NomC, AdresseC, NomF}} (\textbf{Fournisseurs} \bowtie \sigma_{\text{NomP = 'briques'}} (\textbf{Prix} \bowtie (\textbf{Commandes} \bowtie \textbf{Clients})))$

Exercice 2

1. Formuler les requêtes suivantes en langage algébrique et en SQL :

(a) Intervenants de la pièce "L'avare" ?

```
SELECT i.Intervenant
FROM Intervenants i
WHERE i.Titre = 'L avare';
```

$\pi_{\text{Intervenant}} (\sigma_{\text{Titre = 'L avare'}} (\textbf{Intervenants}))$

(b) Intervenants qui n'interviennent pas dans la pièce "L'avare" ?

```
SELECT i.Intervenant
FROM Intervenants i
WHERE i.Intervenant NOT IN
     (SELECT Intervenant
     FROM Intervenants
     WHERE Titre = 'L avare');
```

$\pi_{\text{Intervenant}} (\textbf{Intervenants} - (\sigma_{\text{Titre = 'L avare'}} (\textbf{Intervenants})))$

(c) Intervenants qui sont présents au moins dans un spectacle chaque semaine pendant la saison ?

```
SELECT i.Intervenant, COUNT(DISTINCT s.Semaine)
FROM Intervenants i
JOIN Spectacles s
ON i.Titre = s.Titre
GROUP BY i.Intervenant
HAVING COUNT(DISTINCT s.Semaine) = 4
```

$\pi_{\text{Intervenant}}\,(\sigma_{\text{Semaine} = 4}\,(\,{}_{\text{Intervenant}}\boldsymbol{g}_{\textbf{count}(\text{Semaine})}\,(\textbf{Intervenants} \bowtie \textbf{Spectacles})\,))$

$\boldsymbol{g}$: opérateur d'agrégation

L'algèbre relationnelle est basée sur la théorie des ensembles. Par définition, il n'y a pas de doublon dans un ensemble : ses éléments sont distincts. Le langage SQL ne respecte pas complètement le modèle relationnel, puisqu'il autorise plusieurs *tuples* identiques (sauf lorsqu'une contrainte d'unicité est explicitement appliquée). D'où l'existence de la commande `DISTINCT`.

(d) Noms des salles libres au moins une semaine dans la saison ?

```
SELECT p.Salle, COUNT(DISTINCT p.Semaine)
FROM Places p
GROUP BY p.Salle
HAVING COUNT(DISTINCT p.Semaine) < 4;
```

$\pi_{\text{Salle}}\,(\sigma_{\text{Semaine} < 4}\,(\,{}_{\text{Salle}}\boldsymbol{g}_{\textbf{count}(\text{Semaine})}\,\textbf{Places}\,))$

(e) A quelle date (semaine et jour) reste-t-il des places pour aller voir l'intervenant Dupont (et accessoirement dans quel titre) ?

```
SELECT i.Intervenant, s.Titre, s.Semaine, p.Jour, p.Disponibilite
FROM Places p
JOIN Spectacles s
ON p.Semaine = s.Semaine
AND p.Salle = s.Salle
JOIN Intervenants i
ON i.Titre = s.Titre
WHERE i.Intervenant = 'Paul' /* Choisir ici l'intervenant */
AND p.Disponibilite IS NOT NULL

/* implicit join */
SELECT i.Intervenant, s.Titre, s.Semaine, p.Jour,
p.Disponibilite
FROM Places p, Spectacles s, Intervenants i
WHERE p.Semaine = s.Semaine
AND p.Salle = s.Salle
AND i.Titre = s.Titre
AND i.Intervenant = 'Paul' /* Choisir ici l'intervenant */
AND p.Disponibilite IS NOT NULL
```

$$\pi_{\text{Interv, Titre, Semaine, Jour, Dispo}} ((\text{Spectacles} \bowtie \sigma_{\text{Intervenant = 'Paul'}} (\text{Intervenants})) \bowtie \sigma_{\text{Dispo} \neq \text{NULL}} (\text{Places}))$$

(f) Intervenants qui interviennent dans toutes les salles au cours de la saison ?

```
SELECT i.Intervenant, COUNT(DISTINCT s.Salle)
FROM Intervenants i
JOIN Spectacles s
ON i.Titre = s.Titre
GROUP BY i.Intervenant
HAVING COUNT(DISTINCT s.Salle) = 3
```

$$\pi_{\text{Intervenant}} (\sigma_{\text{Salle} = 3} ({}_{\text{Intervenant}}\boldsymbol{g}_{\textbf{count}(\text{Salle})} (\text{Intervenants} \bowtie \text{Spectacles})))$$