

Conservatoire National des Arts et Métiers

292 Rue St Martin 75141 Paris Cedex 03

Informatique - CNAM, Paris

**Bases de données relationnelles
NFP 107 et NFP 107J**

Exercices dirigés

C. Crochepeyre, M. Ferecatu, P. Rigaux, V. Thion et N. Travers

Solutions

2 mai 2011

Chapitre 1

Conception

1.1 Interprétation de schémas entité/association

1.1.1 Centre médical

On vous donne un schémas E/A (figure 1.1) représentant des visites dans un centre médical. Répondez aux questions suivantes **en fonction des caractéristiques de ce schéma** (i.e. : indiquez si la situation décrite est représentable, indépendamment de sa vraisemblance).

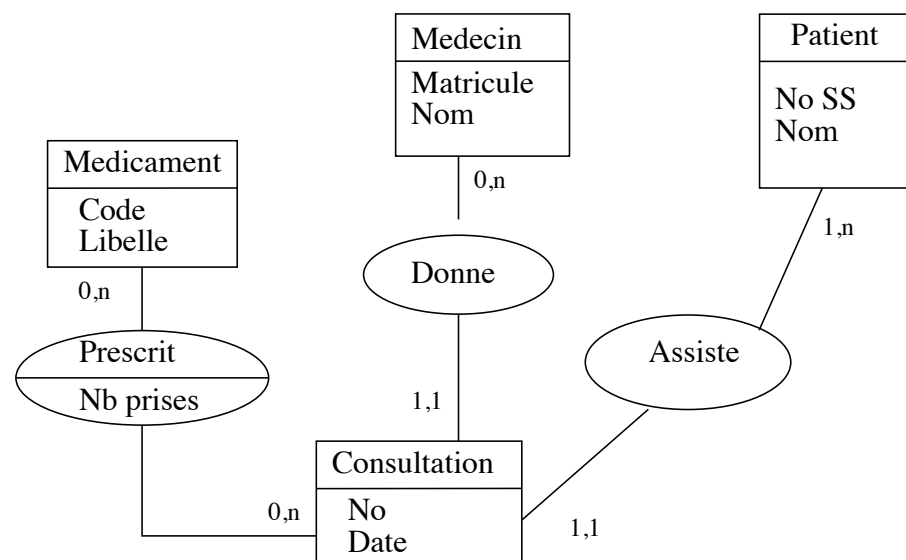


FIGURE 1.1 – Centre médical

Exercice A : Un patient peut-il effectuer plusieurs visites ?

Exercice B : Un médecin peut-il recevoir plusieurs patients dans la même consultation ?

Exercice C : Peut-on prescrire plusieurs médicaments dans une même consultation ?

Exercice D : Deux médecins différents peuvent-ils prescrire le même médicament ?

Solution :

1. Bien sûr.
2. Non (un patient par consultation).
3. Oui.
4. Oui (pas de rapport entre un médecin et une consultation).

1.1.2 Tournoi de tennis

Le second schéma (figure 1.2) représente des rencontres dans un tournoi de tennis.

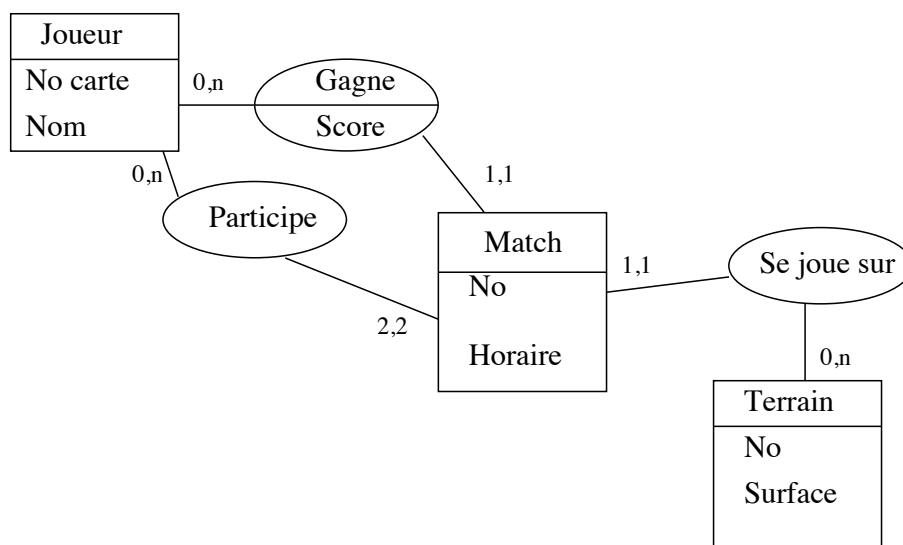


FIGURE 1.2 – Tournoi de tennis

Exercice A : Peut-on jouer des matchs de double ?

Exercice B : Un joueur peut-il gagner un match sans y avoir participé ?

Exercice C : Peut-il y avoir deux matchs sur le même terrain à la même heure ?

Solution :

1. Non (cardinalité 2,2)
2. Oui (pas de lien participe / Gagne)
3. Oui (pas de contrainte sur l'horaire entre 2 matchs)

1.1.3 Un journal

Pour vous entraîner : voici le schéma E/A (figure 1.3 du système d'information (très simplifié) d'un quotidien.

Exercice A : Un article peut-il être rédigé par plusieurs journalistes ?

Exercice B : Un article peut-il être publié plusieurs fois dans le même numéro ?

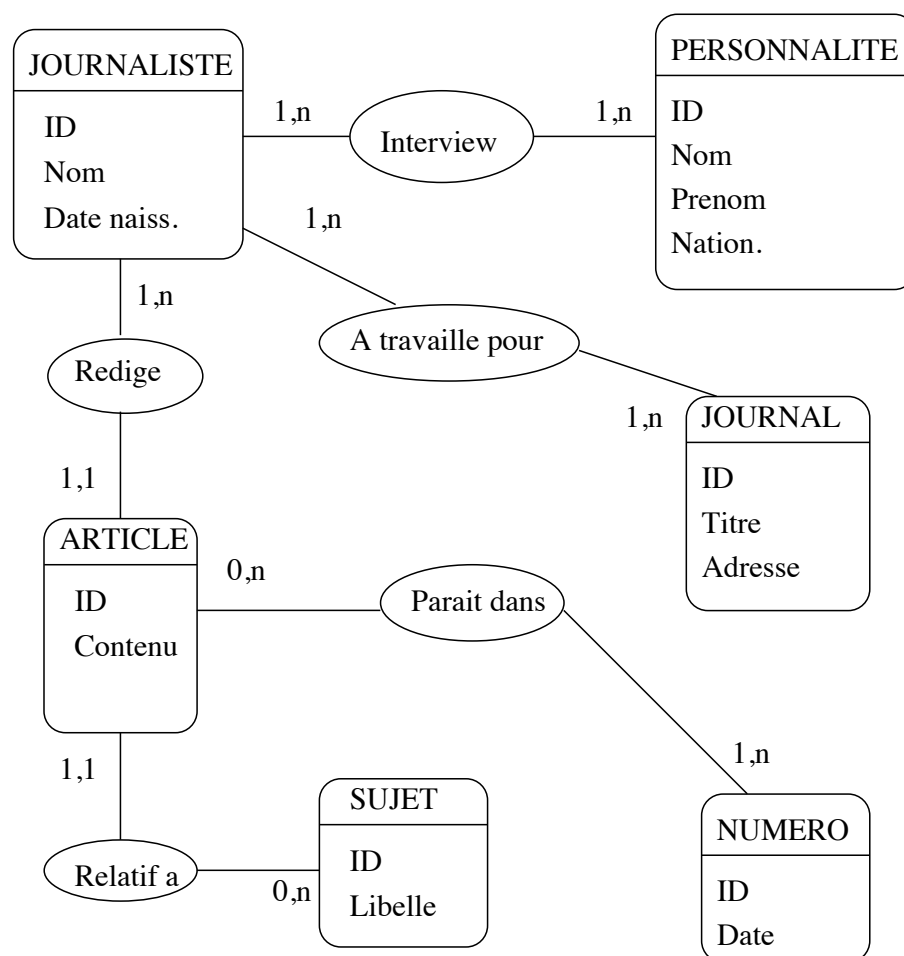


FIGURE 1.3 – Journal

Exercice C : Peut-il y avoir plusieurs articles sur le même sujet dans le même numéro ?

Solution :

1. Non (1,1)
2. Non (lien unique)
3. Oui (pas de lien d'unicité numéro / sujet)

1.2 Modèle relationnel (rappel cycle A)

Exercice A : Pour chacun des schémas E/A donnés précédemment, construire le schéma relationnel correspondant. Indiquez précisément :

- La clé primaire.
- Les clés étrangères.
- Les contraintes éventuelles.

Solution :

Exemple pour le centre médical :

- Médicament (**Code**, Libellé)
- Consultation (**ID-consultation**, Matricule, No-SS, Date). Matricule et No-SS sont les clés étrangères
- Prescription (**Code-médicament**, **ID-consultation**, Nb-prises)
- Médecin (**Matricule**, Nom)
- Patient (**No-SS**, Nom)

Exemple pour le tournoi de tennis :

- Joueur(**No-carte**, Nom)
- Terrain(**No-terrain**, Surface)
- Match(**No-match**, Horaire, No-participant1, No-participant2, No-gagnant, Score, No-terrain). No-participant1, No-participant2, No-gagnant et No-terrain sont clés étrangères.

Exemple pour le journal :

- Journal(**IDJournal**, titre, adresse)
- Journaliste (**IDJournaliste**, Nom, DateNaissance)
- Personnalite (**IDPersonnalite**, Nom, Prénom, Nationalite)
- Interview (**IDJournaliste**, **IDPersonnalite**)
- Article (**IDArticle**, **IDJournaliste**, **IDSujet**, Contenu)
- Sujet (**IDSujet**, Libelle)
- Numéro (**IDNuméro**, Date)
- ParaitDans (**IDArticle**, **IDNuméro**)
- Travail (**IDJournaliste**, **IDJournal**)

Exercice B : Donnez la commande **Create Table** pour les tables *Consultation* et *Match*.

Solution :

Exemple pour la table Consultation :

Create Table Consultation

```
(Id-consultation  NUMBER(10),
Matricule        NUMBER(10),
NO-SS            NUMBER(10),
Date-consultation  DATE,
PRIMARY KEY (Id-consultation),
FOREIGN KEY Matricule REFERENCES Médecin,
FOREIGN KEY NO-SS REFERENCES Patient)
```

1.3 Rétro-conception

On trouve dans un SGBD relationnel les relations ci-dessous. Les clés primaires sont soulignées, mais pas les clés étrangères.

IMMEUBLE (Adresse, Nb-étages, Date-construction, Nom-Gérant)

APPART (Adresse, Numéro, Type, Superficie, Etage)

PERSONNE (Nom, Age, Code-Profession)

OCCUPANT (Adresse, Numéro-Appart, Nom-Occupant, Date-arrivée, Date-départ)

PROPRIÉTÉ (Adresse, Nom-Propriétaire, Quote-part)

TYPE-APPART (Code, Libellé)

PROFESSION (Code, Libellé)

Exercice A : Identifier les clés étrangères dans chaque relation.

Exercice B : Reconstruire le schéma E/A.

Exercice C : Existe-t-il des contraintes d'intégrité ? Lesquelles ?

Exercice D : Certaines données du schéma relationnel résultent-elles d'optimisation ?

Chapitre 2

Algèbre Relationnelle

2.1 Sélection et Projection

Soit la relation

PERSONNE		
Nom	Age	Ville
Marc	29	Paris
Catherine	32	Lyon
Sophie	54	Paris
Claude	13	Montpellier
Serge	40	Lyon

Exercice A : Donnez les résultats des requêtes suivantes :

Requête 1 : $\sigma_{Age=30}(PERSONNE)$ (sélection)

Requête 2 : $\pi_{Age}(PERSONNE)$ (projection)

Requête 3 : $\pi_{Age}(\sigma_{Nom='Serge'}(PERSONNE))$ (projection, sélection)

Exercice B : Exprimez les requêtes suivantes en algèbre relationnelle :

Requête 1 : les personnes (nom, âge, ville) qui habitent Paris.

Solution :

$\sigma_{Ville='Paris'}(PERSONNE)$

Requête 2 : les personnes (nom, âge, ville) qui ont moins de 30 ans.

Solution :

$\sigma_{Age<30}(PERSONNE)$

Requête 3 : les villes dans la relation PERSONNE.

Solution :

$\pi_{Ville}(PERSONNE)$

Requête 4 : les noms des personnes habitant à Paris.

Solution :

$\pi_{Nom}(\sigma_{Ville='Paris'}(PERSONNE))$

2.2 Jointure relationnelle

Exercice A : Soient **R** et **S** les relations

R		S	
A	B	B	C
a	b	b	c
a	f	e	a
c	b	b	d
d	e	g	b

où les attributs A, B, C sont définis sur les domaines des lettres de l'alphabet.
Donnez le résultat des requêtes suivantes :

Requête 1 : $R \bowtie S$ (jointure naturelle).

Requête 2 : $\sigma_{A=C}(\rho_{B/B'}(R) \times S)$ (équi-jointure).

Requête 3 : $R \bowtie S = \pi_R(R \bowtie S)$ (semijoin).

Solution :

$R \bowtie S$			$\sigma_{A=C}(\rho_{B/B'}(R) \times S)$				$R \bowtie S$	
A	B	C	A	B'	B	C	A	B
a	b	c	a	b	e	a	a	b
a	b	d	a	f	e	a	c	b
c	b	c	c	b	b	c	d	e
c	b	d	d	e	b	d		
d	e	a						

Exercice B : Est-ce que les équations suivantes sont vraies ?

$$\pi_{A,B}(R \bowtie S) = R \quad (2.1)$$

$$\pi_{B,C}(R \bowtie S) = S \quad (2.2)$$

Solution :

NON :

$\pi_{A,B}(R \bowtie S)$		$\pi_{B,C}(R \bowtie S)$	
A	B	B	C
a	b	b	c
c	b	b	d
d	e	e	a

2.3 Auto-Jointure et Renommage

Soit **T**(A,B) une relation où A et B prennent leurs valeurs dans le même domaine. Supposons qu'on veuille construire une relation **TS** contenant les seuls n-uplets $\langle a, b \rangle$ tels que $\langle b, a \rangle$ est également un n-uplet de **T**. Par exemple :

T		TS	
A	B	A	B
a	b	a	b
a	c	a	c
c	a	c	a
c	d	b	a
b	a		
e	f		

Exprimez cette opération par une expression de l'algèbre relationnelle.

Solution :

Solution 1 :

1. On fait une copie de **T** dans $S(A,B)$ $S := T$
2. On renomme l'attribut A en A_1 et B en B_1 $S := \rho_{A/A_1, B/B_1}(S)$
3. S a maintenant pour schéma $S(A_1, B_1)$
4. Le résultat est $\mathbf{TS} = T \bowtie_{B=A_1 \wedge A=B_1} S$

Solution 2 :

$$\mathbf{TS} = T \cap \rho_{B/A, A/B}(\mathbf{T})$$

Chapitre 3

Algèbre - SQL : Employés - Départements

3.1 Schéma

Les exemples suivants sont tirés des sources de la société Oracle.

3.1.1 Relation des Employés (EMP)

EMP(ENO, ENOM, PROF, DATEEMB, SAL, COMM, DNO)

ENO : numéro d'employé, clé

ENOM : nom de l'employé

PROF : profession (directeur n'est pas une profession)

DATEEMB : date d'embauche

SAL : salaire

COMM : commission (un employé peut ne pas avoir de commission)

DNO : numéro de département auquel appartient l'employé

3.1.2 Relation des Départements (DEPT)

DEPT(DNO, DNOM, DIR, VILLE)

DNO : numéro de département, clé

DNOM : nom du département

DIR : directeur du département

VILLE : lieu du département (ville)

3.2 Opérations Algébriques

Soit l'exemple suivant :

EMP	ENO	ENOM	PROF	DATEEMB	SAL	COMM	DNO
	10	Joe	Ingénieur	1.10.93	4000	3000	3
	20	Jack	Technicien	1.5.88	3000	2000	2
	30	Jim	Vendeur	1.3.80	5000	5000	1
	40	Lucy	Ingénieur	1.3.80	5000	5000	3

DEPT	DNO	DNOM	DIR	VILLE
	1	Commercial	30	New York
	2	Production	20	Houston
	3	Développement	40	Boston

Exercice A : Calculer $\sigma_{sal < 5000}(EMP)$.

Solution :

<i>ENO</i>	<i>ENOM</i>	<i>PROF</i>	<i>DATEEMB</i>	<i>SAL</i>	<i>COMM</i>	<i>DNO</i>
10	Joe	Ingénieur	1.10.93	4000	3000	3
20	Jack	Technicien	1.5.88	3000	2000	2

Exercice B : Calculer $EMPbis = \rho_{ENO/ENO'}(\pi_{ENO,COMM}(EMP))$

Solution :

EMPbis	<i>ENO'</i>	<i>COMM</i>
	10	3000
	20	2000
	30	5000
	40	5000

Exercice C : Calculer $\pi_{ENO,SAL}(EMP) \bowtie_{SAL=COMM} (EMPbis)$

Solution :

<i>ENO</i>	<i>SAL</i>	<i>ENO'</i>	<i>COMM'</i>
20	3000	10	3000
30	5000	30	5000
40	5000	40	5000
30	5000	40	5000
40	5000	30	5000

Exercice D : Exprimer par une phrase ce qu'on obtient en évaluant les requêtes précédentes.

Solution :

- Expression 1 : on obtient les employés dont le salaire est inférieur à 5000.
- Expression 2 : on obtient le numéro et la commission des employés.
- Expression 3 : on obtient les couples de numéros d'employés dont le premier a un salaire qui est égal à la commission du deuxième.

Exercice E : Quelle est l'expression de l'algèbre relationnelle qui permettrait d'obtenir le nom et la profession de l'employé de numéro 10.

Solution :

$$\pi_{ENOM,PROF}(\sigma_{ENO=10}(EMP))$$

Exercice F : Idem pour la liste des noms des employés qui travaillent à New York.

Solution :

$$\pi_{ENOM}(EMP \bowtie (\sigma_{VILLE='NewYork'}(DEPT)))$$

Exercice G : Idem pour avoir le nom du directeur du département “Commercial”.

Solution :

$$\pi_{ENOM}(EMP \bowtie_{ENO=DIR} \pi_{DIR}(\sigma_{DNOM='Commercial'}(DEPT)))$$

3.3 Requêtes

- Exprimer les requêtes Q1 à Q18 à l’aide de l’algèbre relationnelle.
- Exprimer en SQL les requêtes Q1 à Q24.

3.3.1 Interrogation d’une seule Relation

Requête 1 : Donner tous les n-uplets de DEPT.

Solution :

Algèbre : $DEPT$

SQL :

```
SELECT * FROM DEPT;
```

Requête 2 : Donner tous les n-uplets de EMP.

Solution :

Algèbre : EMP

SQL :

```
SELECT * FROM EMP;
```

Requête 3 : Donner les noms et les salaires des employés.

Solution :

Algèbre : $\pi_{ENOM,SAL}(EMP)$

SQL :

```
SELECT ENOM, SAL
FROM EMP;
```

Requête 4 : Donner les professions des employés (après élimination des duplicats).

Solution :

Algèbre : $\pi_{PROF}(EMP)$

SQL :

```
SELECT DISTINCT PROF
FROM EMP;
```

Requête 5 : Donner les dates d’embauche des techniciens.

Solution :

Algèbre : $\pi_{DATEEMB}(\sigma_{PROF='TECHNICIEN'}(EMP))$

SQL :

```
SELECT DATEEMB
FROM EMP
WHERE PROF='TECHNICIEN';
```

3.3.2 Jointures

Requête 6 : Faire le produit cartésien entre EMP et DEPT.

Solution :

Algèbre : $EMP \times DEPT$

SQL :

```
SELECT *
FROM EMP, DEPT;
```

Requête 7 : Donner les noms des employés et les noms de leur département.

Solution :

Algèbre : $\pi_{ENOM, DNOM}(EMP \bowtie DEPT)$

SQL :

```
SELECT ENOM, DNOM
FROM EMP, DEPT
WHERE EMP.DNO=DEPT.DNO;
```

Requête 8 : Donner les numéros des employés travaillant à BOSTON.

Solution :

Algèbre : $\pi_{ENO}(EMP \bowtie \sigma_{VILLE='BOSTON'}(DEPT))$

SQL :

```
SELECT ENO
FROM EMP, DEPT
WHERE EMP.DNO=DEPT.DNO AND VILLE='BOSTON';
```

Requête 9 : Donner les noms des directeurs des départements 1 et 3. Attention : directeur n'est pas une profession !

Solution :

Algèbre : $\pi_{ENOM}(\sigma_{DNO=1 \vee DNO=3}(DEPT) \bowtie_{DIR=ENO} EMP)$

SQL :

```
SELECT ENOM
FROM EMP, DEPT
WHERE (DEPT.DNO=1 OR DEPT.DNO=3) AND DIR = ENO;
```

ou

```
SELECT ENOM
FROM EMP, DEPT
WHERE DEPT.DNO IN (1,3) AND DIR = ENO;
```

Requête 10 : Donner les noms des employés travaillant dans un département avec au moins un ingénieur.

Solution :

Algèbre :

$$R1 := \pi_{DNO}(\sigma_{PROF='INGENIEUR'}(EMP))$$

$$R2 := \pi_{ENOM}(EMP \bowtie R1)$$

SQL :

```
SELECT E2.ENOM
FROM EMP E1, EMP E2
WHERE E1.DNO = E2.DNO
AND E1.PROF = 'INGÉNIEUR';
```

Requête 11 : Donner le salaire et le nom des employés gagnant plus qu'un (au moins un) ingénieur.

Solution :

Algèbre :

$$\begin{aligned} R1 &:= \rho_{SAL/SAL1}(\pi_{SAL}(\sigma_{PROF='INGENIEUR'}(EMP))) \\ R2 &:= \pi_{ENOM,SAL}(EMP \bowtie_{SAL > SAL1} R1) \end{aligned}$$

SQL :

```
SELECT E1.ENOM, E1.SAL
FROM EMP E1, EMP E2
WHERE E2.PROF='INGENIEUR'
AND E1.SAL > E2.SAL;
```

ou

```
SELECT ENOM, SAL FROM EMP
WHERE SAL > ANY (SELECT SAL
FROM EMP
WHERE PROF='INGENIEUR');
```

Requête 12 : Donner le salaire et le nom des employés gagnant plus que **tous les ingénieurs**.

Solution :

SQL :

```
SELECT ENOM, SAL FROM EMP
WHERE SAL > ALL (SELECT SAL
FROM EMP
WHERE PROF='INGENIEUR');
```

Requête 13 : Donner les noms des employés et les noms de leurs directeurs.

Solution :

Algèbre :

$$\begin{aligned} R1 &:= \rho_{ENOM/DIRNOM}(\pi_{ENOM,DNO}(EMP \bowtie_{ENO=DIR} DEPT)) \\ R2 &:= \pi_{ENOM,DNO} EMP \\ R3 &:= \pi_{ENOM,DIRNOM}(R1 \bowtie R2) \end{aligned}$$

- $R1(DIRNOM,DNO)$: les départements avec les noms de leur directeur
- $R2(ENOM,DNO)$: les employés avec les numéros de leur département
- $R3(ENOM,DIRNOM)$: les employés (ENOM) avec les noms de leur directeur

SQL :

```
SELECT E1.ENOM, E2.ENOM
FROM EMP E1, EMP E2, DEPT D
WHERE E1.DNO=D.DNO AND E2.ENO = D.DIR;
```

Requête 14 : Trouver les noms des employés ayant le même directeur que JIM. Attention : un employé peut être directeur de plusieurs départements.

Solution :

Algèbre :

$$\begin{aligned} R1 &:= \pi_{DIR}(\sigma_{ENOM='JIM'}(EMP) \bowtie DEPT) \\ R2 &:= DEPT \bowtie R1 \\ R3 &:= \pi_{ENOM}(\sigma_{ENOM \neq 'JIM'}(EMP) \bowtie R2) \end{aligned}$$

- $R1(DIR)$: le numéro du directeur de JIM

- $R2(\dots)$: les départements avec le même directeur
- $R3(ENOM)$: les noms des employés ayant le même directeur que JIM.

SQL :

```
SELECT ENOM
FROM EMP
WHERE ENOM <> 'JIM'
      AND DNO IN (SELECT D2.DNO
                  FROM EMP,
                  DEPT D1, DEPT D2
                  WHERE ENOM='JIM'
                  AND D1.DNO = EMP.DNO
                  AND D1.DIR = D2.DIR);
```

Requête 15 : Donner le nom et la date d'embauche des employés embauchés avant leur directeur ; donner également le nom et la date d'embauche de leur directeur.

Solution :

Algèbre :

```
R1 :=  $\pi_{DNO, ENOM, DATEEMB}(EMP \bowtie_{DIR=ENO} DEPT)$ 
R2 :=  $\rho_{ENOM/DIRNOM, DATEEMB/DIRDATE}(R1)$ 
R3 :=  $\pi_{ENOM, DATEEMB, DIRNOM, DIRDATE}(\sigma_{DIRDATE < DATEEMB}(EMP \bowtie R2))$ 
```

- $R1(DNO, ENOM, DATEEMB)$: le nom et la date d'embauche du directeur du dept. DNO.
- $R2(DNO, DIRNOM, DIRDATE)$: renommage des attributs
- $R3(ENOM, DATEEMB, DIRNOM, DIRDATE)$: résultat

SQL :

```
SELECT E1.ENOM, E1.DATEEMB, E2.ENOM, E2.DATEEMB
FROM EMP E1, EMP E2, DEPT D
WHERE E2.ENO=D.DIR
      AND E1.DNO=D.DNO
      AND E1.DATEEMB<E2.DATEEMB;
```

Requête 16 : Donner les départements qui n'ont pas d'employés.

Solution :

Algèbre : $DEPT - (DEPT \bowtie EMP)$

SQL :

```
SELECT *
FROM DEPT
WHERE DNO NOT IN (SELECT DNO FROM EMP);
```

Requête 17 : Donner les noms des employés du département COMMERCIAL embauchés le même jour qu'un employé du département PRODUCTION.

Solution :

Algèbre :

```
R1 :=  $\pi_{DATEEMB}(EMP \bowtie \sigma_{DNOM='PRODUCTION'}(DEPT))$ 
R2 :=  $\pi_{ENOM}((EMP \bowtie \sigma_{DNOM='COMMERCIAL'}(DEPT)) \bowtie R1)$ 
```

SQL :

```
SELECT DISTINCT ENOM
FROM EMP E1, DEPT D1, EMP E2, DEPT D2
WHERE E1.DNO=D1.DNO
      AND E2.DNO=D2.DNO
      AND D1.DNOM='COMMERCIAL'
```



```
AND D2.DNOM='PRODUCTION'
AND E1.DATEEMB=E2.DATEEMB
```

ou

```
SELECT ENOM
FROM EMP, DEPT
WHERE EMP.DNO=DEPT.DNO
AND DNOM='COMMERCIAL'
AND DATEEMB IN (SELECT DATEEMB
FROM EMP, DEPT
WHERE EMP.DNO=DEPT.DNO
AND DNOM='PRODUCTION');
```

Requête 18 : Donner les noms des employés embauchés avant *tous* les employés du département 1.

Solution :

Algèbre :

$$R1 := \rho_{DATEEMB/DATE1}(\pi_{DATEEMB}(\sigma_{DNO=1}(EMP)))$$

$$R2 := \pi_{ENOM}(EMP - (EMP \bowtie_{DATEEMB \geq DATE1} R1))$$

SQL :

```
SELECT ENOM
FROM EMP
WHERE DATEEMB < ALL (SELECT DATEEMB
FROM EMP
WHERE DNO=1);
```

Requête 19 : Donner les noms des employés ayant le même emploi et le même directeur que JOE.

Solution :

Algèbre :

$$R1 := \pi_{DIR, PROF}(\sigma_{ENOM='JOE'}(EMP) \bowtie DEPT)$$

$$R2 := \pi_{ENOM}((EMP \bowtie DEPT) \bowtie R1)$$

- $R1(DIR, PROF)$: le directeur de Joe et sa profession
- $R2(ENOM)$: résultat

SQL :

```
SELECT ENOM
FROM EMP, DEPT
WHERE ENOM <> 'JOE'
AND EMP.DNO = DEPT.DNO
AND (PROF, DIR) = (SELECT PROF, DIR
FROM EMP, DEPT
WHERE ENOM='JOE'
AND EMP.DNO = DEPT.DNO);
```

3.3.3 Valeurs Nulles, Tris, Groupes, Agrégats et Expressions

Requête 20 : Donner la liste des employés ayant une commission.

Solution :

```
SELECT *
FROM EMP
WHERE COMM IS NOT NULL;
```

Requête 21 : Donner les noms, emplois et salaires des employés par emploi croissant et, pour chaque emploi, par salaire décroissant.

Solution :

```
SELECT ENOM, PROF, SAL
FROM EMP
ORDER BY PROF ASC, SAL DESC;
```

Requête 22 : Donner le salaire moyen des employés.

Solution :

```
SELECT AVG(SAL) AS 'SALAIRE MOYEN'
FROM EMP;
```

Requête 23 : Donner le nombre d'employés du département PRODUCTION.

Solution :

```
SELECT COUNT(EMP.*)
FROM EMP, DEPT
WHERE EMP.DNO = DEPT.DNO
AND DEPT.DNOM = 'PRODUCTION';
```

Requête 24 : Les numéros de département et leur salaire maximum ?

Solution :

```
SELECT DNO, MAX(SAL)
FROM EMP
GROUP BY DNO;
```

Requête 25 : Donner les noms des employés ayant le salaire maximum de chaque département.

Solution :

```
SELECT ENOM
FROM EMP
WHERE (DNO, SAL) IN (SELECT DNO, MAX(SAL)
FROM EMP
GROUP BY DNO);
```

ou

```
SELECT ENOM
FROM EMP E
WHERE SAL = (SELECT MAX(SAL)
FROM EMP F
WHERE F.DNO = E.DNO);
```

Requête 26 : Les professions et leur salaire moyen ?

Solution :

```
SELECT PROF, AVG(SAL)
FROM EMP
GROUP BY PROF;
```

Requête 27 : Le salaire moyen le plus bas (par profession) ?

Solution :

```
SELECT MIN (AVG (SAL) )  
FROM EMP  
GROUP BY PROF;
```

Requête 28 : Donner les emplois ayant le salaire moyen le plus bas ; donnez aussi leur salaire moyen.

Solution :

```
SELECT PROF, AVG (SAL)  
FROM EMP  
GROUP BY PROF  
HAVING AVG (SAL) = (SELECT MIN (AVG (SAL) )  
FROM EMP  
GROUP BY PROF) ;
```

Chapitre 4

Algèbre - SQL : Appartements - Écoles

4.1 Schéma

IMMEUBLE (ADI, NBETAGES, DATEC, PROP)
APPIM (ADI, NAPR, OCCUP, TYPE, SUPER, ETAGE)
PERSONNE (NOM, AGE, PROF, ADR, NAPR)
ÉCOLE (NOME, ADEC, NBCLASSES, DIR)
CLASSE (NOME, NCL, MAITRE, NBEL)
ENFANT (NOMP, PRENOM, AN, NOME, NCL)

avec la signification suivante :

1. Relation IMMEUBLE

ADI : adresse d'immeuble, clé ; on fait l'hypothèse pour simplifier, que l'adresse identifie de manière unique un immeuble

NBETAGES : nombre d'étages d'un immeuble

DATEC : date de construction

PROP : nom du propriétaire de l'immeuble qui est une personne

2. Relation APPIM (Appartement)

ADI : adresse d'immeuble

NAPR : numéro d'appartement

OCCUP : occupant de l'appartement (nom de la personne)

TYPE : type de l'appartement (Studio, F2, ...)

SUPER : superficie de l'appartement

ETAGE : étage où se situe l'appartement

3. Relation PERSONNE

NOM : nom de personne, clé ; on fait l'hypothèse pour simplifier, que ce nom est unique sur l'ensemble des personnes que l'on considère dans la base

AGE : âge de la personne

PROF : profession de la personne (directeur d'école n'est pas une profession)

ADR : adresse de la résidence d'une personne, il s'agit d'un immeuble

NAPR : numéro d'appartement

4. Relation ÉCOLE

NOME : nom d'une école, clé

ADEC : adresse d'une école

NBCLASSES : nombre de classes

DIR : nom du directeur

5. Relation CLASSE

NOME : nom d'une école

NCL : nom de la classe, e.g., CP1, CE2, CE3, etc...

MAITRE : nom de l'instituteur

NBEL : nombre d'élèves dans la classe

6. Relation ENFANT

NOMP : nom de la personne responsable de l'enfant, clé e.g., père, mère etc...

PRENOM : prénom de l'enfant

AN : année de naissance

NOME : nom d'une école

NCL : nom de la classe

La relation **IMMEUBLE** décrit un ensemble d'immeubles. Chaque immeuble a un propriétaire. La relation **APPIM** décrit pour chaque immeuble l'ensemble des appartements qui le compose. Chaque appartement peut héberger plusieurs personnes mais il y en a une qui est responsable (par exemple le locataire) et qui est désignée par le constituant **OCCUP**. Si l'appartement est inoccupé, ce constituant prend la valeur **NULL**. La relation **PERSONNE** décrit un ensemble de personnes. **ADR** et **NAPR** représentent l'adresse où réside une personne. Une personne peut avoir plusieurs enfants décrits par la relation **ENFANT**. Pour simplifier, on ne considère que les enfants allant à l'école primaire. Les écoles et les classes sont décrites dans les relations **ÉCOLE** et **CLASSE**. On suppose ici que toutes ces relations vérifient les contraintes d'intégrité de base. Par exemple, une personne répertoriée dans la relation **CLASSE** en tant que maître exerce bien la profession d'instituteur dans la relation **PERSONNE**. De même, un directeur d'école est un instituteur.

4.2 Requêtes

Exprimer les requêtes suivantes à l'aide de l'algèbre relationnelle, puis les traduire en SQL.

Requête 1 : Donner l'adresse des immeubles ayant plus de 10 étages et construits avant 1970.

Solution :

$$\pi_{ADI}(\sigma_{NBETAGES > 10 \wedge DATEC < 1970} IMMEUBLE)$$

```
SELECT ADI
FROM IMMEUBLE
WHERE NBETAGES > 10 AND DATEC < 1970
```

Requête 2 : Donner les noms des personnes qui habitent dans un immeuble dont ils sont propriétaires (occupants et habitants).

Solution :

$$\pi_{NOM}(PERSONNE \bowtie_{NOM=PROP \wedge ADR=ADI} IMMEUBLE)$$

```

SELECT NOM
  FROM PERSONNE, IMMEUBLE
 WHERE NOM = PROP AND ADR = ADI

```

Requête 3 : Donner les noms des personnes qui ne sont pas propriétaires.

Solution :

$$\pi_{NOM}(PERSONNE) - \rho_{PROP/NOM}(\pi_{PROP}(IMMEUBLE))$$

```

SELECT NOM
  FROM PERSONNE
MINUS
SELECT PROP
  FROM IMMEUBLE

```

ou

```

SELECT NOM
  FROM PERSONNE
 WHERE NOM NOT IN (SELECT PROP
                   FROM IMMEUBLE)

```

Requête 4 : Donner les adresses des immeubles possédés par des informaticiens dont l'âge est inférieur à 40 ans.

Solution :

$$R1 := \sigma_{AGE < 40 \wedge PROF = 'INFORMATICIEN'} PERSONNE$$

$$R2 := \pi_{ADI}(R1 \bowtie_{NOM=PROP} IMMEUBLE)$$

```

SELECT ADI
  FROM PERSONNE, IMMEUBLE
 WHERE NOM = PROP AND AGE < 40 AND PROF = 'INFORMATICIEN'

```

Requête 5 : Donner la liste des occupants (nom, âge, profession) des immeubles possédés par DUPONT.

Solution :

$$\pi_{NOM,AGE,PROF}(\sigma_{PROP='DUPONT'}(IMMEUBLE) \bowtie_{ADI=ADR}(PERSONNE))$$

```

SELECT P.NOM, P.AGE, P.PROF
  FROM PERSONNE P, IMMEUBLE I, APPIM A
 WHERE I.ADI = A.ADI
       AND I.PROP = 'DUPONT'
       AND P.NOM = A.OCCUP

```

Requête 6 : Donner le nom et la profession des propriétaires d'immeubles où il y a des appartements vides.

Solution :

$$\begin{aligned} R1 &:= \pi_{ADI, NAPR}(APPIM \bowtie_{OCCUP=NOM} PERSONNE) \\ R2 &:= \pi_{ADI, NAPR} APPIM - R1 \\ R3 &:= \pi_{NOM, PROF}(PERSONNE \bowtie_{NOM=PROP} (IMMEUBLE \bowtie R2)) \end{aligned}$$

- $R1(ADI, NAPR)$: adresses et numéros d'appartement occupés
 - $R2(ADI, NAPR)$: adresses et numéros d'appartement vides
 - $R3(NOM, PROF)$: nom et profession des propriétaires d'immeubles avec des appartements vides
- Avec valeur nulle :

```
SELECT DISTINCT P.NOM, P.PROF
  FROM APPIM A, IMMEUBLE I, PERSONNE P
 WHERE P.NOM = I.PROP AND I.ADI = A.ADI
        AND A.OCCUP IS NULL
```

Sans valeurs nulles :

```
SELECT DISTINCT P.NOM, P.PROF
  FROM APPIM A, IMMEUBLE I, PERSONNE P
 WHERE P.NOM = I.PROP
        AND I.ADI = A.ADI
        AND NOT EXISTS (SELECT *
                        FROM PERSONNE O
                       WHERE O.ADR = I.ADI
                              AND O.NAPR = A.NAPR)
```

Requête 7 : Donner les noms des maîtres qui habitent dans le même immeuble (à la même adresse) qu'au moins un de leurs élèves (on suppose que les enfants vivent sous le même toit que leur responsable).

Solution :

$$\begin{aligned} R1 &:= \pi_{ADR, NOME, NCL, MAITRE}(PERSONNE \bowtie_{MAITRE=NOM} CLASSE) \\ R2 &:= \pi_{ADR, NOME, NCL}(PERSONNE \bowtie_{NOMP=NOM} ENFANT) \\ R3 &:= \pi_{MAITRE}(R1 \bowtie R2) \end{aligned}$$

- $R1(...)$: adresses des maîtres, écoles, classes et maîtres
 - $R2(...)$: adresses des élèves, écoles, classes
 - $R3(MAITRE)$: résultat
- ou :

$$\begin{aligned} R1 &:= \pi_{ADR, MAITRE}(PERSONNE \bowtie_{NOMP=NOM} (ENFANT \bowtie CLASSE)) \\ R2 &:= \pi_{MAITRE}(\sigma_{NOM=MAITRE}(R1 \bowtie PERSONNE)) \end{aligned}$$

- $R1(ADR, MAITRE)$: adresses des enfants et leurs maîtres
- $R2(MAITRE)$: résultat

```
SELECT DISTINCT M.NOM
  FROM CLASSE C, PERSONNE M,
        ENFANT E, PERSONNE R
 WHERE C.MAITRE = M.NOM
```

```

AND E.NOMEC = C.NOMEC
AND E.NCL = C.NCL
AND E.NOMP = R.NOM
AND M.ADR = R.ADR

```

ou (imbriqué) :

```

SELECT C.MAITRE
FROM CLASSE C, PERSONNE P
WHERE C.MAITRE = P.NOM AND P.ADR IN (SELECT P.ADR
                                     FROM PERSONNE P, ENFANT E
                                     WHERE P.NOM = E.NOMP AND
                                           E.NOMEC = C.NOMEC AND
                                           E.NCL = C.NCL)

```

Requête 8 : Donner l'adresse de l'immeuble, la date de construction, le type d'appartement et l'étage où habitent chacun des maîtres des enfants de DUPONT.

Solution :

```

R1 :=  $\pi_{MAITRE}(\sigma_{NOMP='DUPONT'} ENFANT \bowtie CLASSE)$ 
R2 :=  $R1 \bowtie_{MAITRE=NOM} PERSONNE$ 
R3 :=  $\pi_{ADI,DATEC,TYPE,ETAGE}(R2 \bowtie IMMEUBLE)$ 

```

- $R1(MAITRE)$: les maîtres des enfants de DUPONT
- $R2(...)$: les noms, adresses, ... des maîtres
- $R3(ADI,DATEC,TYPE,ETAGE)$: résultat

```

SELECT A.ADI, I.DATEC, A.TYPE, A.ETAGE
FROM CLASSE C, ENFANT E, PERSONNE P, IMMEUBLE I, APPIM A
WHERE I.ADI = P.ADR
      AND A.NAPR = P.NAPR
      AND A.ADI = I.ADI
      AND P.NOM = C.MAITRE
      AND C.NOMEC = E.NOMEC
      AND C.NCL = E.NCL
      AND E.NOMP = 'DUPONT'

```

Requête 9 : Donner le nom et l'âge des maîtres qui habitent dans un immeuble dont le propriétaire est responsable d'un de leurs élèves.

Solution :

```

R1 :=  $\pi_{MAITRE,AGE,ADR,NOMP}(ENFANT \bowtie (CLASSE \bowtie_{MAITRE=NOM} PERSONNE))$ 
R2 :=  $\pi_{MAITRE,AGE}(R1 \bowtie_{NOMP=PROP \wedge ADR=ADI} IMMEUBLE)$ 

```

- $R1(MAITRE,AGE,ADR,NOMP)$: pour chaque enfant : le nom, l'âge et l'adresse des maîtres et le nom du responsable
- $R2(MAITRE,AGE)$: résultat

```

SELECT M.NOM, M.AGE
FROM IMMEUBLE I, ENFANT E, CLASSE C, PERSONNE M
WHERE I.ADI = M.ADR
      AND I.PROP = E.NOMP
      AND C.NCL = E.NCL
      AND C.NOMEC = E.NOMEC
      AND M.NOM = C.MAITRE

```


Requête 10 : Donner le nom et l'âge des personnes qui sont propriétaires mais qui ne sont ni maître ni directeur d'école (des classes et écoles répertoriées dans ECOLE et CLASSE).

Solution :

$$\begin{aligned}
 R1 &:= \rho_{MAITRE/PROP}(\pi_{MAITRE} CLASSE) \\
 R2 &:= \rho_{DIR/PROP}(\pi_{DIR} ECOLES) \\
 R3 &:= \pi_{PROP} IMMEUBLE - (R1 \cup R2) \\
 R4 &:= \pi_{AGE,NOM}(PERSONNE \bowtie_{PROP=NOM} R3)
 \end{aligned}$$

```

SELECT NOM, AGE
FROM PERSONNE
WHERE NOM IN (SELECT PROP
               FROM IMMEUBLE
               MINUS
               (SELECT DIR
                FROM ECOLE
                UNION
                SELECT MAITRE
                FROM CLASSE))

```

4.3 Mise à jour

Requête 11 : Ajouter un enfant de nom **np**, de prénom **e**, né en **a** et l'inscrire à la classe **c** de l'école **ec**.

Solution :

```

INSERT INTO ENFANT VALUE (np, e, a, ec, c);
UPDATE CLASSE
SET NBEL = NBEL + 1
WHERE NOME = ec AND NCL = c;

```

4.4 Contraintes

Indiquer de la façon la plus formelle possible certaines contraintes que les données de la base doivent respecter pour être conformes à la réalité modélisée ici.

Solution :

Inclusions des ensembles, par exemple :

- $IMMEUBLE[PROP] \subseteq PERSONNE[NOM]$
- $ECOLE[DIR] \subseteq PERSONNE[NOM]$
- $CLASSE[MAITRE] \subseteq PERSONNE[NOM]$
- $ENFANT[NOMP] \subseteq PERSONNE[NOM]$

L'étage d'un appartement dans un immeuble est inférieur ou égal au nombre d'étages de cet immeuble :

$$(IMMEUBLE(a, n, d, p,) \wedge APPIM(a, ap, o, t, s, e)) \Rightarrow e \leq n$$

Le nombre d'élèves dans une classe correspond à l'ensemble des n -uplets dans la relation ENFANT :

$$CLASSE(e, c, m, n) \Rightarrow \text{card}(\{ENFANT : NOME C = e \wedge NCL = c\}) = n$$

Pour les professions

$$\forall d \in \{ECOLE[DIR]\} \Rightarrow \exists a, ad, ap PERSONNE(d, a, 'INSTITUTEUR', ad, ap)$$

$$\forall m \in \{CLASSE[MAITRE]\} \Rightarrow \exists a, ad, ap PERSONNE(m, a, 'INSTITUTEUR', ad, ap)$$

etc...

Chapitre 5

SQL - Algèbre : Cinémas - Films

5.1 Schéma

Les exemples suivants sont tirés du livre *Foundations of Databases* de S. Abiteboul, R. Hull et V. Vianu.

SALLE (Nom, Horaire, Titre)
FILM (Titre, Realisateur, Acteur)
PRODUIT (Producteur, Titre)
VU (Spectateur, Titre)
AIME (Spectateur, Titre)

Un film est réalisé par un metteur en scène mais peut être financé par plusieurs Producteurs. Un Spectateur peut aimer un film sans l’avoir vu.

5.2 Requêtes

Écrire les requêtes suivantes en algèbre relationnelle .

5.2.1 Interrogation d’une seule relation

Requête 1 : Dans quelle salle et à quelle heure peut on voir le film “Mad Max” ?

Solution :

Algèbre :

$$\pi_{Nom, Horaire}(\sigma_{Titre='Mad Max'}(SALLE))$$

SQL :

```
SELECT Nom, Horaire
FROM SALLE
WHERE Titre = 'Mad Max'
```

Requête 2 : Quels sont les films réalisés par Orson Welles ?

Solution :

Algèbre :

$$\pi_{Titre}(\sigma_{Realisateur='Welles'}(FILM))$$

SQL :

```
SELECT Titre
FROM FILM
WHERE Realisateur = 'Welles'
```

Requête 3 : Quels sont les Acteurs du film “Ran” ?

Solution :

Algèbre :

$$\pi_{Acteur}(\sigma_{Titre='Ran'}(FILM))$$

SQL :

```
SELECT Acteur
FROM FILM
WHERE Titre = 'Ran'
```

5.2.2 Jointures

Requête 4 : Dans quelles salles peut-on voir un film avec Simone Signoret ?

Solution :

Algèbre :

$$\pi_{Nom}(SALLE \bowtie (\sigma_{Acteur='Signoret'}(FILM)))$$

SQL :

```
SELECT Nom
FROM SALLE, FILM
WHERE SALLE.Titre = FILM.Titre
AND FILM.Acteur = 'Signoret'
```

Requête 5 : Dans quelles salles peut on voir Marlon Brando après 16h ?

Solution :

Algèbre :

$$\pi_{Nom}(\sigma_{Horaire>16}(SALLE) \bowtie (\sigma_{Acteur='Brando'}(FILM)))$$

SQL :

```
SELECT Nom
FROM SALLE, FILM
WHERE SALLE.Titre = FILM.Titre
AND FILM.Acteur = 'Brando'
AND SALLE.Horaire > 16
```

Requête 6 : Quels sont les Acteurs qui ont produit un film ?

Solution :

Algèbre :

$$\pi_{Acteur}(FILM \bowtie_{Acteur=Producteur} PRODUIT)$$

SQL :

```
SELECT Acteur
  FROM PRODUIT, FILM
 WHERE Acteur = Producteur
```

Requête 7 : Quels sont les Acteurs qui ont produit un film dans lequel ils jouent ?

Solution :

Algèbre :

$$\pi_{Acteur}(\sigma_{Acteur=Producteur}(FILM \bowtie PRODUIT))$$

SQL :

```
SELECT Acteur
  FROM FILM, PRODUIT
 WHERE FILM.Titre = PRODUIT.Titre
       AND Acteur = Producteur
```

Requête 8 : Quels sont les Acteurs qui ont produit et réalisé un même film ?

Solution :

Algèbre :

$$R1 := \pi_{Realisateur}(\sigma_{Producteur=Realisateur}(PRODUIT \bowtie FILM))$$

$$R2 := \pi_{Acteur}(FILM \bowtie_{Acteur=Realisateur} R1)$$

SQL :

```
SELECT A.Acteur
  FROM FILM A, FILM B, PRODUIT C
 WHERE A.Acteur = B.Realisateur
       AND B.Realisateur = C.Producteur
       AND B.Titre = C.Titre
```

Requête 9 : Quels sont les Producteurs qui regardent les films qu'ils ont produits ?

Solution :

Algèbre :

$$\pi_{Producteur}(\sigma_{Spectateur=Producteur}(PRODUIT \bowtie VU))$$

5.2.3 Difference

Requête 10 : Quels films ne passent en ce moment dans aucune salle ?

Solution :

Algèbre :

$$\pi_{Titre}(FILM) - \pi_{Titre}(SALLE)$$

Requête 11 : Quels Spectateurs aiment un film qu'ils n'ont pas vu ?

Solution :

Algèbre :

$$\pi_{Spectateur}(AIME - VU)$$

Requête 12 : Qui n'aime aucun film qu'il a vu ?

Solution :

Algèbre :

$$\pi_{Spectateur}(VU) - \pi_{Spectateur}(AIME \bowtie VU)$$

Requête 13 : Qui n'a produit aucun film de Doillon ?

Solution :

Algèbre :

$$\pi_{Producteur}(PRODUIT) - \pi_{Producteur}(PRODUIT \bowtie (\sigma_{Realisateur='Doillon'} FILM))$$

Requête 14 : Qui a produit un film qui ne passe dans aucune salle ?

Solution :

Algèbre :

$$\pi_{Producteur}(PRODUIT \bowtie (\pi_{Titre} FILM - \pi_{Titre} SALLE))$$

5.2.4 Division

Requête 15 : Quels Spectateurs ont vu tous les films ? (ou Spectateurs pour lesquels il n'existe pas un film qu'ils n'ont pas vu)

Solution :

Algèbre :

$$VU \div \pi_{Titre}(FILM)$$

SQL :

```
SELECT Spectateur
FROM Vu v1
WHERE NOT EXISTS ( SELECT *
                    FROM FILM f
                    WHERE NOT EXISTS ( SELECT *
                                      FROM Vu v2
                                      WHERE v2.Titre = f.Titre
                                      AND v2.Spectateur = v1.Spectateur
                                      ) )
```

Requête 16 : Quels Acteurs jouent dans tous les films de Welles ? (ou Acteurs pour lesquels il n'existe pas un film de Welles qu'ils n'ont pas joué)

Solution :

Algèbre :

$$(\pi_{Acteur,Titre}(FILM)) \div (\pi_{Titre}(\sigma_{Realisateur='Welles'}(FILM)))$$

SQL :

```
SELECT fx.Acteur
FROM FILM fx
WHERE NOT EXISTS ( SELECT fy
                    FROM FILM fy
                    WHERE fy.Realisateur = 'Welles'
                    AND NOT EXISTS ( SELECT fz
                                      FROM FILM fz
                                      WHERE fz.Titre = fy.Titre
                                      AND fz.Acteur = fx.Acteur
                                      ) )
```

Requête 17 : Quels sont les Spectateurs qui aiment tous les films qu'ils ont vu ? (ou Spectateurs pour lesquels il n'existe pas un film qu'ils ont vu et qu'ils n'ont pas aimé)

Solution :

Algèbre :

$$(AIME \bowtie \rho_{Spectateur/Spectateur'}(VU)) \div VU$$

Requête 18 : Quels sont les Producteurs qui voient tous les films qu'ils ont produit ? (ou Producteurs pour lesquels il n'existe pas un film qu'ils ont produit et qu'ils n'ont pas vu)

Solution :

Algèbre :

$$(\pi_{Producteur} PRODUIT \bowtie_{Producteur=Spectateur} VU) \div PRODUIT$$

Requête 19 : Quels Producteurs voient tous les films de Kurosawa ? (ou Producteurs pour lesquels il n'existe pas un film de Kurosawa qu'ils n'ont pas vu)

Solution :

Algèbre :

$$\begin{aligned} R1 &:= \pi_{Titre}(\sigma_{Realisateur='Kurosawa'} FILM) \\ R2 &:= VU \bowtie_{Spectateur=Producteur} PRODUIT \\ R3 &:= \pi_{Spectateur}(R2 \div R1) \end{aligned}$$

- $R1(Titre)$: tous les Titres de films de Kurosawa
- $R2(\dots)$: les films vus par des Producteurs
- $R3(Spectateur)$: résultat