

## R2.06 Exploitation BDD

M. T. Pham, M. Khayata

[minh-tan.pham@univ-ubs.fr](mailto:minh-tan.pham@univ-ubs.fr)

Cours 3

# Division

# Plan

## Compréhension de la division

Division normale (inexacte)

Division exacte

Division : mapping par différence

Division : mapping par comptage (lecture complémentaire)

# Division

- Soient R1 et R2 deux relations telles que tous les attributs de R2 se retrouvent dans R1
- La division **R1 / R2** est une relation ayant seulement les attributs de R1 non présents dans R2.
- Les t-uples de **R1 / R2** apparaissant dans R1 doivent en plus contenir dans R1 tous les t-uples de R2.

**R1**

A	X
a1	x1
a2	x2
a1	x3
a1	x2
a2	x1

**R2**

X
x1
x2

**R1/R2**

A
a1
a2

# Division

**Voiture**

immat	marque	puissance	dept
24ET7898	Renault	8	22
76YU9087	Peugeot	8	56
75GY6435	Audi	8	35
67HR4321	Peugeot	7	35
46FC5687	Renault	7	56
55YT9462	Peugeot	9	22

**Moto**

id	marque	puiss
34E87	Yamaha	8
87Y54	Yamaha	7
98I09	Honda	8

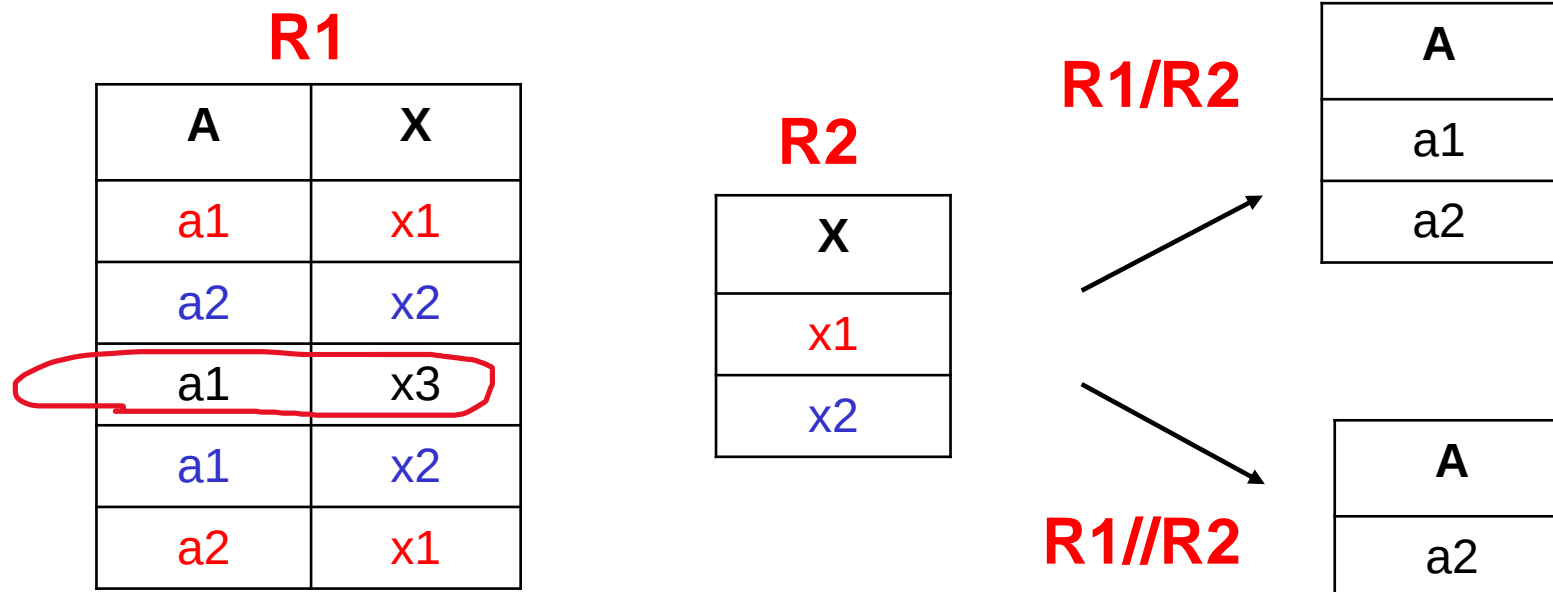
Quelles sont les marques des voitures dont les puissances sont celles de toutes les motos ?

**Division:** *Voiture[marque,puissance] / Moto[puiss]*

marque
Renault
Peugeot

# Division exacte

- Soient R1 et R2 deux relations telles que tous les attributs de R2 se retrouvent dans R1
- La **division exacte**  $R1 // R2$  est une relation ayant seulement les attributs de R1 non présents dans R2.
- Les t-uples de  $R1 // R2$  apparaissant dans R1 doivent en plus contenir dans R1 tous les t-uples de R2 et uniquement ceux-là.



# Division exacte

**Voiture**

immat	marque	puissance	dept
24ET7898	Renault	8	22
76YU9087	Peugeot	8	56
75GY6435	Audi	8	35
67HR4321	Peugeot	7	35
46FC5687	Renault	7	56
55YT9462	Peugeot	9	22

**Moto**

id	marque	puiss
34E87	Yamaha	8
87Y54	Yamaha	7
98I09	Honda	8

*Quelles sont les marques des voitures dont les puissances sont celles de toutes les motos et uniquement celles-là (et pas d'autres) ?*

**Division exacte :** *Voiture[marque,puissance] // Moto[puiss]*

marque
Renault

# Division - SQL

- N'existe pas de façon directe !!!
- On est obligé de procéder :
  - soit à travers une différence (**MINUS**, ou **EXCEPT** avec MySQL)
  - soit à travers un comptage (**COUNT**) → *lecture complémentaire*
- Nous allons traiter la division avec le mapping par différence.

# Plan

## Compréhension de la division

Division normale (inexacte)

Division exacte

## Division : mapping par différence

Division : mapping par comptage (lecture complémentaire)



# Division normale

**Voiture**

immat	marque	puissance	dept
24ET7898	Renault	8	22
76YU9087	Peugeot	8	56
75GY6435	Audi	8	35
67HR4321	Peugeot	7	35
46FC5687	Renault	7	56
55YT9462	Peugeot	9	22

**Moto**

id	marque	puiss
34E87	Yamaha	8
87Y54	Yamaha	7
98I09	Honda	8

Quelles sont les marques des voitures dont les puissances sont celles de toutes les motos ?

**Division:** *Voiture[marque,puissance] / Moto[puiss]*

marque
Renault
Peugeot

# Division normale

*Quelles sont les marques des voitures dont les puissances sont celles de toutes les motos ?*

- On cherche donc les **marques de voitures** telles que si on regarde les **puissances associées dans la table voiture** on retrouve **toutes les puissances des motos**.
- En prenant l'ensemble des puissances de toutes les motos, et en faisant la différence avec les puissances associée aux marques cherchées dans la table voiture **on doit retrouver l'ensemble vide**.
- C'est ce que nous allons mettre en œuvre !

# Division normale

*Quelles sont les marques des voitures dont les puissances sont celles de toutes les motos ?*

L'ensemble des puissances des motos :

```
SELECT DISTINCT puiss  
FROM Moto;
```

La différence avec les puissances d'une marque de voiture :

```
MINUS  
SELECT DISTINCT puissance  
FROM Voiture  
WHERE marque = « une marque donnée »;
```

# Division normale

*Quelles sont les marques des voitures dont les puissances sont celles de toutes les motos ?*

La requête finale est :

```
SELECT DISTINCT marque
FROM Voiture V1
WHERE NOT EXISTS ( SELECT puiss
                     FROM Moto
                     MINUS
                     SELECT DISTINCT puissance
                     FROM Voiture V2
                     WHERE V2.marque = V1.marque
                   ) ;
```

# Division exacte

**Voiture**

immat	marque	puissance	dept
24ET7898	Renault	8	22
76YU9087	Peugeot	8	56
75GY6435	Audi	8	35
67HR4321	Peugeot	7	35
46FC5687	Renault	7	56
55YT9462	Peugeot	9	22

**Moto**

id	marque	puiss
34E87	Yamaha	8
87Y54	Yamaha	7
98I09	Honda	8

*Quelles sont les marques des voitures dont les puissances sont celles de toutes les motos et uniquement celles-là (et pas d'autres) ?*

**Division exacte :** *Voiture[marque,puissance] // Moto[puiss]*

marque
Renault

# Division exacte

*Quelles sont les marques des voitures dont les puissances sont celles de toutes les motos et uniquement celles-là ?*

- On procède exactement de la même manière que pour la division normale.
- Mais il faut tout simplement rajouter la condition que la différence entre les puissances des marques cherchées et les puissances des motos est vide aussi.
- ce qui donnera au final l'égalité des deux ensembles

# Division exacte

*Quelles sont les marques des voitures dont les puissances sont celles de toutes les motos et uniquement celles-là ?*

```
SELECT DISTINCT marque
FROM Voiture V1
WHERE NOT EXISTS ( SELECT puiss
                    FROM Moto
                    MINUS
                    SELECT DISTINCT puissance
                    FROM Voiture V2
                    WHERE V2.marque = V1.marque)
)
AND NOT EXISTS ( SELECT DISTINCT puissance
                  FROM Voiture V2
                  WHERE V2.marque = V1.marque
                  MINUS
                  SELECT puiss
                  FROM Moto
                  ) ;
```

# Exemple

**Produit** (code(1), libelle)

**ALivrer**([unProduit=@Produit.code, unClient](1))

*Quelles sont les clients auxquels on doit livrer tous les produits dont le libellé commence par 'A' ?* → **ALivrer / Produit{libelle='A%'}[code]**

```
SELECT DISTINCT .....  
FROM ..... A1  
WHERE NOT EXISTS ( SELECT .....  
                     FROM .....  
                     WHERE .....  
                     MINUS  
                     SELECT DISTINCT .....  
                     FROM ..... A2  
                     WHERE A2 . ..... = A1 . .....  
                     ) ;
```



# Exemple

**Produit** (code(1), libelle)

**ALivrer**([unProduit=@Produit.code, unClient](1))

*Quelles sont les clients auxquels on doit livrer tous les produits dont le libellé commence par 'A' ?* → **ALivrer / Produit{libelle='A%'}[code]**

```
SELECT DISTINCT unClient
FROM Alivrer A1
WHERE NOT EXISTS ( SELECT code
                    FROM Produit
                    WHERE libelle LIKE 'A%'
                    MINUS
                    SELECT DISTINCT unProduit
                    FROM Alivrer A2
                    WHERE A2.unClient = A1.unClient
                );
```

# Exemple

**Produit** (code(1), libelle)

**ALivrer**([unProduit=@Produit.code, unClient](1))

*Quelles sont les clients auxquels on doit livrer tous les produits dont le libellé commence par 'A' et uniquement ceux-là ?* → **ALivrer // Produit{libelle='A%'}[code]**

```
SELECT DISTINCT unClient
FROM Alivrer A1
WHERE NOT EXISTS ( SELECT code
                    FROM Produit
                    WHERE libelle LIKE 'A%'
                    MINUS
                    SELECT DISTINCT unProduit
                    FROM Alivrer A2
                    WHERE A2.unClient = A1.unClient);
AND NOT EXISTS ( SELECT DISTINCT unProduit
                  FROM Alivrer
                  WHERE A2.unClient = A1.unClient
                  MINUS
                  SELECT code
                  FROM Produit
                  WHERE libelle LIKE 'A%' );
```

# Exemples pratiques

**Compagnie** (idComp (1), nomComp, pays, estLowCost)

**Pilote** (idPilote(1), nomPilote, nbHVol, compPil=@Compagnie(idComp))

**TypeAvion** (idTypeAvion(1),nbPassagers)

**Qualification** (unPilote=@Pilote(idPilote)(1),unTypeAvion=@TypeAvion(idTypeAvion)(1))

**Avion** (idAvion(1), leTypeAvion=@TypeAvion(idTypeAvion)(NN), compAv=@Compagnie(idComp)(NN) )

## Compagnie

idComp	nomComp	pays	estLowCost
1	Air France	France	0
2	Corsair International	France	0
3	EasyJet	Angleterre	1
4	American Airlines	Etats-Unis	0
5	Ryanair	Irlande	1

## TypeAvion

idTypeAvion	nbPassagers
A320	174
A350	324
B747	279

## Qualification

unPilote	unTypeAvion
1	A320
1	A350
2	A320
2	B747
3	A320
4	A320
4	A350
4	B747
5	A350
5	A320
7	A350
7	B747

## Pilote

idPilote	nomPilote	nbHVol	compPil
1	Ridard	1500	1
2	Naert	450	3
3	Godin	450	5
4	Fleurquin	3000	1
5	Pham	900	4
6	Kerbellec	900	
7	Kamp	3000	4

## Avion

idAvion	leTypeAvion	compAv
1	A320	1
2	A320	3
3	A350	1
4	A320	2
5	B747	1
6	A350	4
7	B747	4
8	A320	5
9	A320	5

# Exemples pratiques

Afficher les noms des pilotes possédant une qualification **pour tous les avions**.



```
SELECT nomPilote
FROM Pilote
WHERE NOT EXISTS
(
  SELECT idTypeAvion
  FROM TypeAvion
  MINUS
  SELECT unTypeAvion
  FROM Qualification
  WHERE unPilote = idPilote
)
```

RESULT

NOMPILOTE
Fleurquin

# Exemples pratiques

Afficher les noms des pilotes possédant une qualification **pour tous les avions de leur compagnie.**



## Solution

```
SELECT nomPilote
FROM Pilote
WHERE NOT EXISTS
(
  SELECT leTypeAvion
  FROM Avion
  WHERE compAv = compPil
  MINUS
  SELECT unTypeAvion
  FROM Qualification
  WHERE unPilote = idPilote
)
AND compPil IS NOT NULL — Pourquoi est-ce nécessaire ?
;
```

## RESULT

NOMPILOTE

---

Naert  
Godin  
Fleurquin  
Kamp

# Exemples pratiques

Afficher les noms des pilotes possédant une qualification **pour exactement tous les avions de leur compagnie.**



## Solution

```
SELECT nomPilote
FROM Pilote
WHERE NOT EXISTS
(
  SELECT leTypeAvion
  FROM Avion
  WHERE compAv = compPil
  MINUS
  SELECT unTypeAvion
  FROM Qualification
  WHERE unPilote = idPilote
)
AND NOT EXISTS
(
  SELECT unTypeAvion
  FROM Qualification
  WHERE unPilote = idPilote
  MINUS
  SELECT leTypeAvion
  FROM Avion
  WHERE compAv = compPil
)
AND compPil IS NOT NULL
;
```

RESULT

NOMPILOTE

Godin  
Fleurquin  
Kamp

# Plan

## Compréhension de la division

Division normale (inexacte)

Division exacte

Division : mapping par différence

**Division : mapping par comptage** (lecture complémentaire)

# Mapping par comptage

**Commerce** (code(1), ville, type)

*Quels sont les villes où tous les types de commerces possibles sont implantés ?*

→ **Commerce[ville, type] / Commerce[type]**

```
SELECT ville
FROM Commerce
GROUP BY ville
HAVING COUNT(DISTINCT type) >= ( SELECT COUNT(DISTINCT type)
                                FROM Commerce
                                ) ;
```

**Remarque : test = pour la division exacte**



# Mapping par comptage - Remarque

**Voiture**

immat	marque	puissance	dept
24ET7898	Renault	8	22
76YU9087	Peugeot	8	56
75GY6435	Audi	8	35
67HR4321	Peugeot	7	35
46FC5687	Renault	7	56
55YT9462	Peugeot	9	22

**Moto**

id	marque	puiss
34E87	Yamaha	8
87Y54	Yamaha	7
98I09	Honda	8

*Quelles sont les marques des voitures dont les puissances sont celles de toutes les motos ?*

Est-ce que ça fonctionne pour cet exemple ?

# Mapping par comptage - Remarque

**Voiture**

immat	marque	puissance	dept
24ET7898	Renault	8	22
76YU9087	Peugeot	8	56
75GY6435	Audi	8	35
67HR4321	Peugeot	7	35
46FC5687	Renault	7	56
55YT9462	Audi	9	22

**Moto**

id	marque	puiss
34E87	Yamaha	8
87Y54	Yamaha	7
98I09	Honda	8

*Quelles sont les marques des voitures dont les puissances sont celles de toutes les motos ?*

**Est-ce que ça fonctionne pour cet exemple ?**

# Mapping par comptage - Remarque

- Pour implémenter une division par un mapping avec comptage et regroupement, la table par laquelle on divise **doit contenir toutes les valeurs possibles**.
- Autrement dit, le domaine de l'attribut X de la table "dividende » (R1) doit être inclus dans celui de l'attribut X de la table "diviseur » (R2).

**R1**

A	X
a1	x1
a2	x2
a3	x1
a2	x1
a2	x2

**R2**

X
x1
x2

**R1**

A	X
a1	x1
a2	x2
a1	x3
a1	x2
a2	x1

**R2**

X
x1
x2

# Références

- <https://fr.wikipedia.org/>
- Cours BDD Anthony Ridard, <https://math-ridard.fr/>
- Christian Soutou, *Modélisation des bases de données : UML et les modèles entité-association*, 3<sup>ème</sup> édition, Groupe Eyrolles.
- Christian Soutou, *SQL pour Oracle : Optimisation des requêtes et schémas*, 5<sup>ème</sup> édition, Groupe Eyrolles.
- Elisabetta De Mria, *Cours L2 Informatique*, UFR Sciences, Université Côte d'Azur, <https://www.i3s.unice.fr/%18edemaria/>
- Oracle SQL Tutorials, <https://www.w3schools.com/sql/default.asp>
- MySQL Tutorials, <https://www.w3schools.com/MySQL/default.asp>